

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE
CONCENTRATION RÉSEAUX DE TÉLÉCOMMUNICATIONS
M.Ing.

PAR
Stéphane DAVY

PERFORMANCE DES RÉSEAUX MAILLÉS MULTIRADIO
SUR BANC DE TEST

MONTRÉAL, LE 12 AOÛT 2010

©Tous droits réservés, Stéphane Davy, 2010

PRÉSENTATION DU JURY
CE MÉMOIRE A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE

M. Zbigniew Dziong, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Jean-Charles Grégoire, codirecteur de mémoire
Énergie, Matériaux et Télécommunications à Institut national de la recherche scientifique

M. Jean-Marc Robert, président du jury
Département de génie logiciel et des T.I. à l'École de technologie supérieure

M. Roch H. Glitho, membre du jury
Département de génie logiciel et des T.I. à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 5 AOÛT 2010

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

J'aimerais tout d'abord remercier mon directeur de recherche, Zbigniew Dziong, professeur au département de génie électrique à l'École de technologie supérieure, pour son encadrement et soutien monétaire tout au long de ce projet. Merci pour avoir organisé et structuré des périodes de présentation au sein du groupe LAGRIT qui m'ont permis d'apprendre à mieux appréhender un auditoire.

J'aimerais spécialement remercier mon co-directeur de recherche, Jean-Charles Grégoire, professeur au département énergie, matériaux et télécommunications à l'Institut national de la recherche scientifique, pour son encadrement et son soutien moral tout au long de ce projet. Merci pour sa disponibilité, son discernement et sa confiance. Et un grand merci pour les stagiaires qu'il m'a en partie confié pendant quatre mois. Une expérience très enrichissante pour moi en encadrement et formation d'équipiers.

J'aimerais ainsi remercier Lionel Huck et Édouard Lavaud les stagiaires qui ont travaillé à mes côtés pendant quatre mois. Ils ont beaucoup apporté à l'initialisation du projet et du développement de la plate-forme. Merci à vous de m'avoir permis de m'essayer à l'encadrement de personnes.

J'aimerais remercier les étudiants des groupes de travail LAGRIT et Co-Mesh pour les échanges informels que nous avons tenus à la fin des présentations d'avancement. Ils m'ont permis d'y voir plus clair dans mon projet et de me soutenir dans mes avancées.

J'aimerais remercier ma mère, Béatrice Riedel pour son soutien moral et financier pendant ces six dernières années. Merci pour m'avoir fait prendre conscience de l'importance de la lecture. Enfin, merci pour ses multiples relectures de mon mémoire qui m'ont permis de l'affiner.

J'aimerais remercier mon père, Jean Davy pour son soutien moral. Merci pour son ouverture d'esprit et sa faculté à trouver des solutions simples et judicieuses aux problèmes.

J'aimerais remercier ma famille, avec une pensée particulière pour ma sœur Marianne Davy et son fils en bas âge Charlie Chaplet. Un grand merci à mes grands-parents pour la sagesse, le calme et la sérénité qu'ils m'ont transmis. Cela m'a beaucoup aidé dans le développement et la mise en place de ce projet.

Enfin, j'aimerais remercier mes amis qui m'ont toujours soutenu dans mes entreprises. Je remercie tout particulièrement Laurent Boivin, un ami originaire du Québec, pour m'avoir fait découvrir la culture québécoise avec autant de précisions et de sincérité.

PERFORMANCE DES RÉSEAUX MAILLÉS MULTIRADIO SUR BANC DE TEST

Stéphane DAVY

RÉSUMÉ

Les réseaux WiFi maillés de niveau 2 sont apparus en 2003 avec la première proposition de normalisation du 802.11s. Au fil des années l'approche de niveau 2 a montré des limites, la recherche s'est alors dirigée vers des réseaux maillés de niveau 3. L'idée des réseaux maillés multiradio a alors émergé; les routeurs sont équipés de deux radios ou plus : une pour l'accès et une ou plus pour le relaying. L'objectif principal de ce mémoire est d'étudier la performance des réseaux WiFi maillés multiradio sur banc de test en ce basant sur l'existant. L'approche sur banc test nous a amené à développer une plate-forme de contrôle de banc de test qui nous a permis d'automatiser l'exécution d'expériences réseau. Enfin, son utilisation nous a amené à réfléchir sur un modèle d'abstraction basé sur le vocabulaire du jeu qui permettrait de rendre l'élaboration d'expérience plus intuitive sur la plate-forme et ainsi de la rendre accessible à un plus large public.

La plate-forme d'automatisation d'expériences sur banc de test se base sur l'utilisation de l'existant. Elle synchronise l'exécution de commandes Unix sur des nœuds distants dans une approche modulaire. Une expérience est définie par un script d'expérience qui contient l'ensemble des lignes de commandes Unix associées aux nœuds. Le script est exécuté par un interpréteur de script qui lit les commandes une à une et les envoie aux nœuds associés. Chaque nœud exécute les commandes du script. Le traitement des données en résultat est lui aussi réalisé par des scripts extérieurs à la plate-forme. Le fonctionnement de la plate-forme s'adapte ainsi à tout type de problématiques de recherche. L'exécution et le traitement des données sont contrôlés par la plate-forme qui retourne automatiquement les résultats à l'utilisateur en fin d'expérience.

L'étude des réseaux maillés multiradio a démontré un haut niveau de performance. Des limites dans la gestion de l'équité ont été relevées au niveau des embranchements et sauts WiFi; ils ont été corrigés par une politique simple de QoS de niveau 3. Une limite en bande passante de 25 Mbps des points d'accès a limité les effets de saturation. Des files d'attente à priorités égales ont amélioré l'équité de 15%. Et enfin, des tunnels ont stabilisé les trafics et permis de contrôler la répartition de la bande passante.

Une nouvelle méthodologie de mise en place de banc de test est proposée pour faciliter l'utilisation de la plate-forme. Un modèle d'abstraction basé sur le vocabulaire du jeu est développé pour ouvrir l'élaboration de scripts d'expériences et de résultats à des utilisateurs sans connaissance Unix. Une expérience réseau est ainsi définie comme une partie de jeu (exécution de l'expérience) dans laquelle des joueurs (nœuds) déplacent (trafics) des pions (outils Unix) sur un plateau (le banc de test réseau) en suivant des règles (simulation de comportement réseau, ex : taux de perte). Pour permettre un accès distant intuitif, le modèle d'abstraction est introduit dans une interface graphique Web.

Mots-clés : wifi maillé, multiradio, performance, équité, TCP, QoS, banc de test, plate-forme de contrôle, exécution automatique, centralisé, modèle d'abstraction, jeu, intuitif.

PERFORMANCE ANALYSIS OF MULTIRADIO WIRELESS MESH NETWORKS ON A TEST BED

Stéphane DAVY

ABSTRACT

Level 2 WiFi Mesh Networks appeared in 2003 with the first proposal for standardization of the IEEE 802.11s. Over the years the level 2-approach has shown limitations from its own and then the research was directed to the study of level 3 mesh networks. Then, the idea of multiradio mesh networks emerged; the routers are equipped with two or more radios: one for the access and one or more for the relay. The main objective of this thesis is to study the performance of multiradio WiFi mesh networks on a test bed, based on the use of existing technologies. The experimental approach has led us to develop a platform to control the test bed and automated the execution of experiments. Finally, its use has led us to reflect on a model of abstraction based on the game vocabulary that would make the use of the platform more intuitive and thus make accessible to a wider audience.

The platform is developed in a modular way to automate the execution of experiences on a test-bed. An experience is defined by a script which synchronizes the execution of UNIX commands on the nodes in the test-bed. Data processing is also done by the use of external scripts. This allows the platform to be customized to any research problem. Execution and data processing are controlled by the platform which automatically returns the results to the user.

The study of multiradio wireless mesh networks demonstrated a high level of performance. Limitations have been identified in the equity management due to WiFi branches and jumps. They have been corrected by the use of a simple level 3 QoS policy. A bandwidth limit of 25 Mbps to the access points has reduced the effects of saturation. Weighted fair queuing has improved the equity of 15%. And finally, tunnels have stabilized traffic and allowed to manage the bandwidth sharing.

A new methodology is proposed to easily set up a test-bed. An abstraction model based on the game vocabulary is developed to open the usage of test-bed to users with no knowledge of UNIX. A network experiment is defined as a game play (experience execution) in which players (nodes) move (traffics) pawns (UNIX tools) on a chessboard (the network test-bed) by following rules (simulation behaviour, e.g. lost rate). To enable remote access, the abstraction model is placed in a Web GUI interface.

Keywords: wireless mesh network, multiradio, performance, fairness, equity, TCP, QoS, test-bed, control platform, centralized, autoexecution, abstraction model, game, intuitive.

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 REVUE DE LA LITTÉRATURE	4
1.1 Réseaux maillés (WMN)	4
1.1.1 Architecture.....	5
1.1.2 Routeurs maillés.....	7
1.1.3 Routage	8
1.1.4 Brouillon IEEE 802.11s	9
1.1.5 Technologies multiradio	9
1.2 Techniques d'évaluation	10
1.2.1 Approches	10
1.2.2 Axes de recherche explorés.....	12
1.2.2.1 Côté réseau	12
1.2.2.2 Côté gestion de trafic.....	13
1.2.2.3 Côté trafics (TCP).....	13
1.2.3 Objectifs	14
1.3 Performance : définition et mesures	15
1.3.1 RTT	15
1.3.2 Bande passante	15
1.3.3 Stabilité	16
1.3.4 Équité	17
1.4 Banc de test : exemple de MiNT	17
1.4.1 Architecture.....	18
1.4.1.1 Nœud de test	19
1.4.1.2 Nœud de contrôle	19
1.4.2 Contrôle d'expérience.....	19
1.4.3 Configuration de la topologie	20
1.4.3.1 Configuration des trafics	20
1.4.3.2 Configuration de la mobilité.....	20
1.4.3.3 Paramètres WiFi	20
1.4.3.4 Exécution de l'expérience	20
1.4.4 Traitement des résultats	21
1.4.4.1 Collection des traces.....	21
1.4.4.2 Agrégation des traces	21
1.4.4.3 Visualisation des résultats	21
1.4.5 Limites et problèmes.....	21
1.5 Conclusion.....	22
CHAPITRE 2 PLATE-FORME DE CONTRÔLE DE BANC DE TEST	24
2.1 Problématique.....	25
2.1.1 Test de performance	25

2.1.2	Réseau	25
2.1.2.1	Éléments de simulation.....	26
2.1.2.2	Trafics.....	26
2.1.2.3	Mesure	26
2.1.2.4	Banc de test.....	26
2.1.2.5	Nœuds.....	27
2.1.2.6	Services.....	27
2.1.3	Approche.....	28
2.1.3.1	Gestion des nœuds.....	28
2.1.3.2	Gestion des services	29
2.1.3.3	Ouvert sur l'existant.....	30
2.2	Expérience	30
2.2.1	Exécution	30
2.2.1.1	Variables.....	31
2.2.1.2	Réseau.....	31
2.2.1.3	Outils de simulation.....	32
2.2.1.4	Mesure et trafics	32
2.2.1.5	Relâchement des nœuds	33
2.2.1.6	Autres	33
2.2.2	Traitement des données.....	35
2.2.2.1	Journal	35
2.2.2.2	Traces	35
2.2.2.3	Données	36
2.2.2.4	Résultats	37
2.3	Architecture	38
2.3.1	Les nœuds	39
2.3.2	Le serveur de contrôle.....	41
2.4	Dessin logiciel	41
2.4.1	Le gestionnaire de nœud	42
2.4.1.1	Caractéristiques d'un nœud.....	43
2.4.1.2	Connexion avec le serveur.....	43
2.4.1.3	Exécution de commandes	44
2.4.2	Le système de contrôle.....	45
2.4.2.1	Contrôle des nœuds	45
2.4.2.2	Gestionnaire d'exécution des commandes	46
2.4.2.3	Contrôle des expériences.....	48
2.4.2.4	Acquisition et extraction des données	48
2.4.2.5	Mise en forme des résultats	49
2.4.3	La base de données	49
2.4.4	L'accès utilisateur.....	50
2.4.4.1	Terminal	50
2.4.4.2	Web.....	51
2.5	Discussion.....	51
2.6	Conclusion.....	52

CHAPITRE 3	PERFORMANCE DES RESEAUX WIFI MAILLES MULTIRADIO ...	53
3.1	Paramétrage	53
3.1.1	Débit.....	54
3.1.2	Puissance de transmission	55
3.1.3	Canaux	55
3.1.4	Interférences.....	56
3.1.5	Synthèse	58
3.2	Performance des réseaux WiFi maillés	59
3.2.1	Effet des sauts	59
3.2.2	Équité, vue d'ensemble.....	60
3.2.3	Internet	63
3.2.4	Taux de perte.....	64
3.2.5	Partage de la bande passante.....	66
3.3	Mécanismes de QoS et performance	67
3.3.1	Limite en bande passante des points d'accès.....	68
3.3.2	Files d'attente	69
3.3.3	Tunnels.....	70
3.3.4	Pertes et mécanismes de QoS	71
3.4	Discussions	71
3.4.1	Limites et observations	71
3.4.2	Mécanismes de QoS.....	72
3.4.3	Exploitation en environnement réel	72
3.5	Conclusion.....	73
CHAPITRE 4	MODÈLE D'ABSTRACTION	74
4.1	Approche	74
4.2	Niveaux.....	75
4.2.1	Pion	75
4.2.2	Joueur.....	77
4.2.3	Plateau.....	78
4.2.4	Règle	78
4.2.5	Déplacement	79
4.2.6	Partie	80
4.2.7	Score	81
4.3	Discussion.....	82
4.4	Conclusion.....	82
CHAPITRE 5	DISCUSSION.....	84
CONCLUSION	87
ANNEXE I	Script d'expérience : performance d'une connexion Ethernet	89
ANNEXE II	Spécification de la carte Atheros MiniPCI AR5413.....	90
ANNEXE III	Statistiques de la plate-forme.....	93

ANNEXE IV	Modèle d'abstraction : script d'expérience.....	94
LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES		96

LISTE DES TABLEAUX

	Page
Tableau 2.1	Caractéristiques carte Alix 3d2.....39
Tableau 2.2	Caractéristiques carte Alix 2d3.....40
Tableau 2.3	Caractéristiques du serveur Dell.....41
Tableau 4.1	Performance des canaux WiFi (Mbps)56
Tableau 4.2	Interférences 802.11a VS 802.11g (Mbps).....57
Tableau 4.3	Interférences inter 802.11a (Mbps).....58
Tableau 4.4	Performance des liens "à vide" (Mbps)59
Tableau 4.5	Impact du nombre de sauts (Mbps)60
Tableau 4.6	Équité VS Perte uniforme (Mbps)65
Tableau 4.7	Équité VS Perte localisée (Mbps).....65
Tableau 4.8	Équité VS Multiplication des trafics (Mbps).....66
Tableau 4.9	Partage de la bande passante dans un point d'accès WiFi (Mbps)67
Tableau 4.10	Impact de la limite en bande passante et des files d'attente (Mbps)70
Tableau 4.11	Équité VS Taille du tunnel (Mbps).....70
Tableau 4.12	Mécanismes de QoS et perte uniforme de 1% (Mbps)71
Tableau 3.1	Modèle d'abstraction75
Tableau 3.2	Pions 1/276
Tableau 3.3	Pions 2/276
Tableau 3.4	Joueurs77
Tableau 3.5	Plateau.....78
Tableau 3.6	Règle79
Tableau 3.7	Déplacement80
Tableau 3.8	Partie81

Tableau 3.9	Score	82
-------------	-------------	----

LISTE DES FIGURES

	Page
Figure 1.1	Réseau maillé (WMN).....5
Figure 1.2	Architectures de réseau maillé.....6
Figure 1.3	Structures de réseau maillé.8
Figure 1.4	Architecture de MiNT.....18
Figure 2.1	Traitement des données : exemple fichier journal.35
Figure 2.2	Traitement des données : exemple fichier trace.36
Figure 2.3	Traitement des données : exemple fichier données.36
Figure 2.4	Traitement des données : débit moyen VS taux de perte.....37
Figure 2.5	Architecture du réseau de contrôle.38
Figure 2.6	Schéma du logiciel.....42
Figure 2.7	Initialisation de la connexion.....44
Figure 2.8	Exécution synchrone.....46
Figure 2.9	Exécution asynchrone avec attente.....47
Figure 2.10	Exécution asynchrone avec destruction.....48
Figure 4.1	Topologie multisaut.54
Figure 4.2	Topologie pour l'étude des interférences.57
Figure 4.3	Topologie pour l'étude des sauts WiFi.....60
Figure 4.4	Trafics ascendants.....61
Figure 4.5	Trafics descendants.....62
Figure 4.6	Trafics simultanés : ascendants et descendants.62
Figure 4.7	Internet : Bande passante limité à 10 Mbps.....63
Figure 4.8	Internet : Délais de 100 ms.64
Figure 4.9	Topologie pour l'étude du partage.....67

Figure 4.10	Topologie arbre.....	68
Figure 4.11	Équité VS limite en bande passante du point d'accès.	69

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

3G	Troisième génération
ACK	Accusé de réception (Acknowledgment)
AP	Point d'accès WiFi (Access Point)
Async	Asynchrone
CSV	Valeurs séparées par des virgules (Comma Separated Values)
D	Brouillon (Draft)
GUI	Interface graphique utilisateur (Graphical User Interface)
HWMP	Protocol de routage hybride pour les réseaux maillés sans fil (Hybrid Wireless Mesh Protocol)
ICMP	Protocole de contrôle de l'Internet (Internet Control Message Protocol)
IEEE	Institute of Electrical and Electronics Engineers
MIB	Base d'information de management (Management Information Base)
MiNT	Banc de test réseau miniaturisé (Miniaturized Network Testbed)
MP	Routeur maillé de relai (Mesh Point)
MPP	Routeur maillé, passerelle (Mesh Portal Point)
MSS	Taille maximal de segment (Maximum Segment Size)
NS	Simulateur réseau (Network Simulator)
PDA	Assistant personnel (Personal Digital Assistant)
QoS	Qualité de service (Quality of Service)
RTS/CTS	Requête d'envoi / Libre d'envoi (Request To Send / Clear To Send)
RTT	Temps d'aller-retour (Round Trip Time)

SACK	Accusé de réception sélectif (Selective Acknowledgment)
SIP	Protocole d'initialisation de session (Session Initiation Protocol)
SNMP	Protocole simple de management de réseau (Simple Network Management Protocol)
SSH	Terminal Shell sécurisé (Secure Shell)
STA	Station WiFi
Sync	Synchrone
TCL	Outil de langage commande (Tool Command Language)
TCP	Protocole de contrôle de transmission (Transmission Control Protocol)
Tx	Taux
TXOP	Occasion de transmission (Transmission Opportunity)
UDP	Protocole de datagrammes utilisateur (User Datagram Protocol)
VoIP	Voix sur IP (Voice over IP)
WAN	Réseau étendu (Wide Area Network)
WMN	Réseau maillé sans fil (Wireless Mesh Network)

LISTE DES SYMBOLES ET UNITÉS DE MESURE

Dbm Puissance, décibels

GHz Fréquence, hertz

Mbps Débit, bit/s

INTRODUCTION

Les réseaux WiFi maillés sont une nouvelle approche dans les réseaux sans fil. La structure maillée permet d'accroître la surface de couverture d'un réseau WiFi tout en conservant ses hauts débits. Ils répondent à une demande croissante des utilisateurs pour des équipements mobiles supportant des applications de plus en plus gourmande en bande passante. Les technologies du WiFi IEEE 802.11 sont envisagées pour leur faible coût technologique et la non-nécessité de licence radio. C'est ainsi que des projets de réseaux WiFi maillés de taille métropolitaine commencent à voir le jour dans plusieurs villes du monde. Les technologies utilisées sont généralement propriétaires, le standard 802.11s n'ayant toujours pas été validé par la communauté.

Les solutions de réseau maillé ont commencé à se développer en 2003 avec la première proposition du brouillon 802.11s. Il était alors envisagé un réseau maillé de niveau 2. Au fil des années cette approche a montré des limites dans sa capacité à gérer la performance. La recherche s'est alors dirigée vers des solutions de réseau maillé de niveau 3 offrant plus de souplesse dans la gestion des trafics et des utilisateurs. D'autre part, grâce à la baisse du coût des technologies radio, des radios ont été ajoutées aux routeurs maillés. C'est ainsi qu'aujourd'hui on parle de réseau maillé multiradio. Une radio est dédiée à l'accès et une ou plus au relaying. Cette évolution apporte de nouvelles perspectives de performance mais aussi son lot de défis.

La performance d'un réseau maillé multiradio dépend de l'environnement (obstacles, murs, bureaux...), du niveau physique (radio, puissance, interférences...), du niveau liaison (MAC, aiguillage, gestion de l'accès au média...), du niveau réseau (routage, QoS...) et du niveau transport (TCP, UDP...). Dans le passé, la recherche s'est concentrée à des études spécifiques de ces paramètres. Des axes d'évolution ont été proposés, mais ils remettent bien souvent en cause les protocoles existants. C'est ainsi que nous proposons une étude sur banc de test de la performance effective des réseaux maillés multiradio en utilisant des équipements existants.

L'étude de la performance sur banc de test est intéressante car l'environnement étudié est quasi réel; les comportements expérimentaux sont étudiés sur des équipements réels. Mais ces études nécessitent un nombre important de manipulations pour la configuration des nœuds et la mise en place des trafics. Il est ainsi nécessaire d'utiliser une plate-forme permettant de synchroniser les nœuds dans une approche banc de test. Des plates-formes ont été développées dans le passé, mais elles se sont limitées à des problématiques de performance spécifiques non transférable à notre étude. C'est ainsi que nous proposons une plate-forme d'automatisation de l'exécution d'expériences réseau sur banc de test basée sur une approche modulaire et ouverte pour répondre à un large éventail de problématiques.

La mise en place de banc de test réseau a bien trop souvent été réservée aux utilisateurs Unix. C'est ainsi que nous proposons un modèle d'abstraction basé sur le vocabulaire du jeu permettant l'élaboration d'expériences dans la plate-forme. Une expérience réseau est ainsi définie comme une partie de jeu (exécution) dans laquelle des joueurs (nœuds) déplacent (trafics) des pions (outils Unix) sur un plateau (réseau) en suivant des règles (simulation de comportement réseau, ex : un taux de perte sur une interface WiFi). L'utilisation de la plate-forme devient ainsi plus intuitive.

C'est ainsi que ce mémoire propose l'étude des perspectives de performance des réseaux maillés multiradio sur banc de test. Le banc de test est contrôlé par une plate-forme créée pour la circonstance. La plate-forme permet de synchroniser les technologies existantes dans une approche modulaire basée sur des scripts. L'interface graphique Web est rendue intuitive par le modèle d'abstraction proposé, basé sur le vocabulaire du jeu.

Ce mémoire se décompose en quatre chapitres. Le premier chapitre est l'étude de la littérature. Il présente les réseaux maillés, leur fonctionnement et architecture. Il fait un tour d'horizon des axes de recherches étudiés. Puis, il présente les critères de performance et outils de mesure. Enfin, il présente l'analyse des limites du système de contrôle de banc de test : MiNT. Le deuxième chapitre présente la plate-forme expérimentale développée. Il présente l'approche choisie, définit la notion d'expérience et présente les principaux mécanismes de la

plate-forme. Le troisième chapitre présente les études de performance des réseaux maillés multiradio. Les limites de performance sont évaluées. Puis, des mécanismes de gestion de trafic simples sont proposés pour améliorer la performance. Le quatrième chapitre propose un modèle d'abstraction basé sur le vocabulaire du jeu pour rendre plus intuitive l'utilisation de la plate-forme. Pour finir, une discussion et une conclusion sont proposées.

CHAPITRE 1

REVUE DE LA LITTÉRATURE

Les réseaux maillés sont en développement depuis les années 2003 en réponse au besoin grandissant des utilisateurs en services Internet sans fil (Ian F. Akyildiz, Xudong Wang et Weilin Wang, 2005). Les technologies maillées se basent sur la norme WiFi IEEE 802.11, qui, contrairement au WiMax ou au 3G ne nécessitent pas de licences onéreuses permettant l'utilisation des bandes de fréquence. Les réseaux WiFi maillés permettent d'étendre les capacités des réseaux WiFi classiques tout en conservant leur faible coût. Des routeurs de relais sont ajoutés autour du point d'accès connecté à Internet pour étendre la zone de couverture du réseau WiFi et accueillir plus d'utilisateurs.

Au fil des années, les technologies de réseau maillé se sont améliorées. Les réseaux supportent plus de routeurs, répartissent mieux la charge et routent plus efficacement le trafic. Les routeurs sont équipés d'au moins deux radios permettant de mieux répartir la charge et de séparer l'accès du relayage. Les problématiques de relais sont isolées des problématiques d'accès. Dans ce mémoire on s'intéresse principalement aux problématiques de relayage dans les réseaux maillés. Les problématiques d'accès ayant déjà été largement étudiées.

Dans ce chapitre sont d'abord présentés les réseaux WiFi maillés, les technologies existantes, leur fonctionnement et leur évolution. Puis, les axes de développement engagés par la recherche sont présentés. Ensuite, la performance est définie et les outils de mesure présentés. Enfin, une plate-forme d'étude de performance sur banc de test est présentée.

1.1 Réseaux maillés (WMN)

Un réseau maillé (Wireless Mesh Network) est un réseau de communication constitué de nœuds radio. Sa topologie est constituée de trois éléments : clients, routeurs maillés et passerelles (Figure 1.1). Les clients sont des équipements mobiles (ordinateur portable,

notebook, PDA...) qui se connectent au réseau maillé. Les routeurs maillés forment le réseau dorsal; ils relaient le trafic entre les clients et la passerelle. La passerelle est l'élément de sortie du réseau, il permet l'accès au réseau Internet.

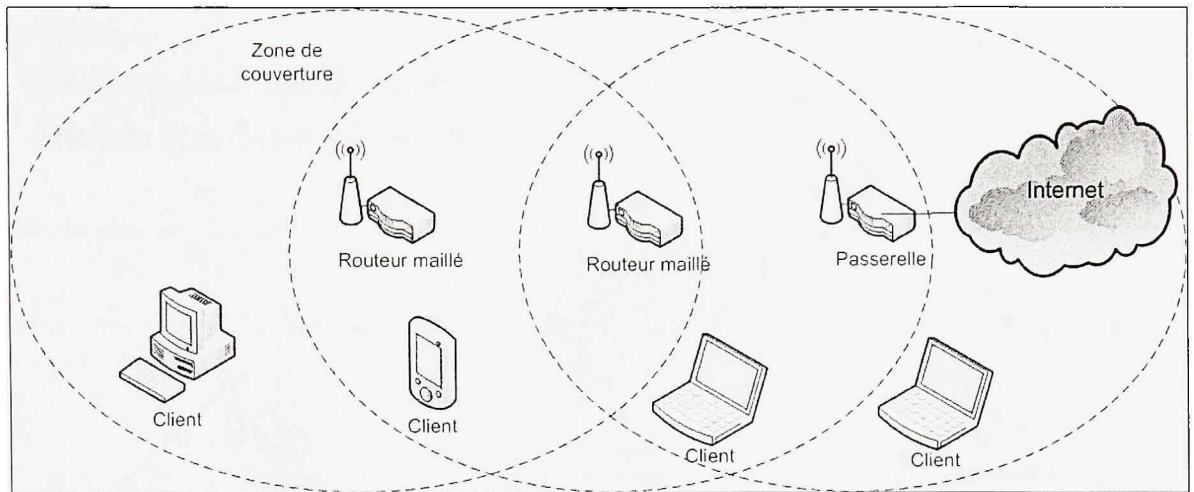


Figure 1.1 Réseau maillé (WMN).

Les réseaux maillés se configurent automatiquement. Lorsqu'un routeur est activé, il scrute l'environnement radio et se connecte au routeur le plus proche. Le réseau met alors à jour ses tables de routages. La topologie est ainsi maintenue à jour après tout changement dans son organisation.

1.1.1 Architecture

Les réseaux WiFi maillés sont des réseaux de routeurs sans fil. Ils permettent à la fois d'offrir un accès sans fil à des clients WiFi et de relayer les données entre ces clients et l'Internet.

Trois architectures de réseaux maillés ont été développées pour répondre aux problématiques d'accès et de relayage (Figure 1.2) (Xudong Wang et Kiyon, 2005) :

- **Infrastructure** : Les routeurs maillés forment un réseau dorsal pour relayer les données entre clients et Internet. Les clients sont accueillis sur le réseau d'accès isolé du réseau dorsal.
- **Client** : Les clients seuls forment le réseau maillé.
- **Hybride** : Les clients et les routeurs maillés forment le réseau maillé.

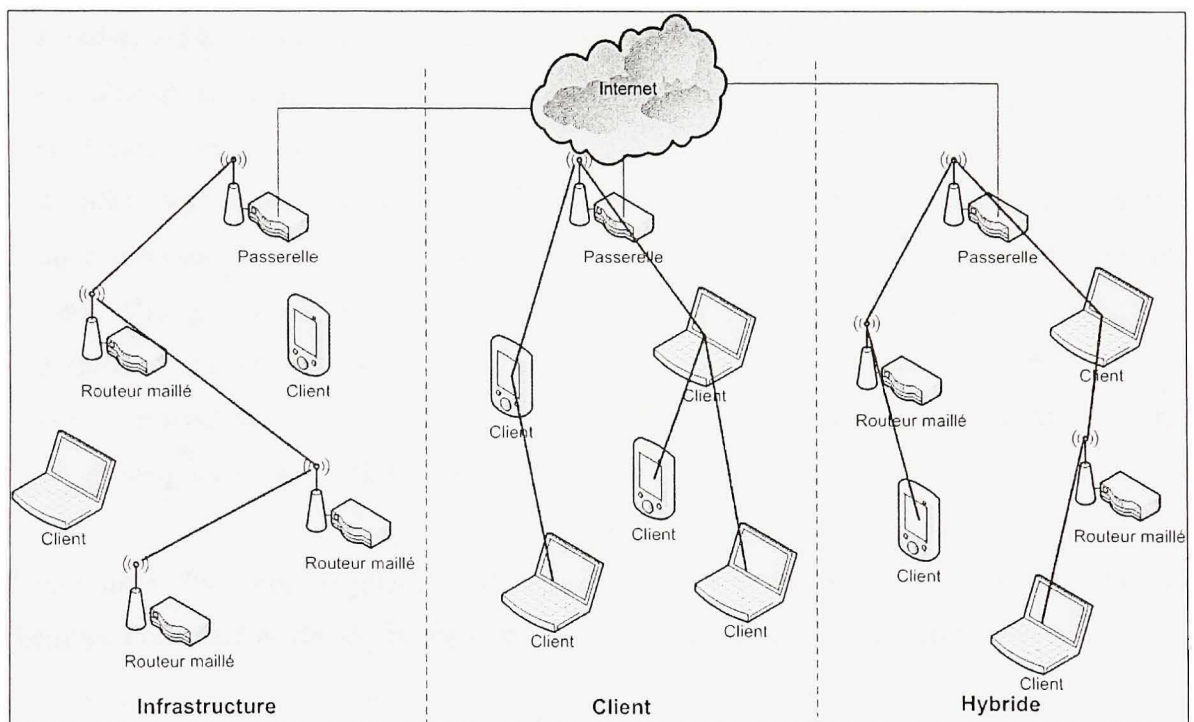


Figure 1.2 Architectures de réseau maillé.

Pour cette étude, l'architecture infrastructure a été retenue. Elle permet une séparation claire des problématiques d'accès des problématiques de relayage en deux réseaux.

1.1.2 Routeurs maillés

Les réseaux maillés de type infrastructure sont partagés en deux réseaux : accès et dorsal. Au fil des années, trois types de routeur maillé ont été développés en se basant sur l'évolution du coût des technologies radio; des radios ont été ajoutées sur les routeurs pour une meilleure répartition de la bande passante.

Il existe aujourd'hui trois types de routeur maillé (Figure 1.3) (BelAir Networks, 2006) :

- **1 radio, 1/2 accès et 1/2 dorsale** : une unique radio est utilisée pour accueillir les clients et transmettre l'information dans le réseau dorsal. La ressource est partagée entre émission et réception des clients et routeurs maillés.
- **2 radios, 1 accès et 1 dorsale** : une radio est utilisée pour accueillir les clients et une autre pour le réseau dorsal. Les deux radios sont sur des canaux sans interférence. Les réseaux sont isolés physiquement pour une meilleure répartition du trafic dans le réseau maillé.
- **3 radios, 1 accès et 2 dorsales** : une radio est utilisée pour accueillir les clients et deux pour le réseau dorsal (connexion « ascendante » vers la passerelle et « descendante » vers les clients). Le réseau dorsal est plus performant.

Il pourrait y avoir encore plus de radios sur le réseau dorsal pour créer une diversité de chemins. Pour cette étude, la structure de routeur maillé à 3 radios a été retenue.

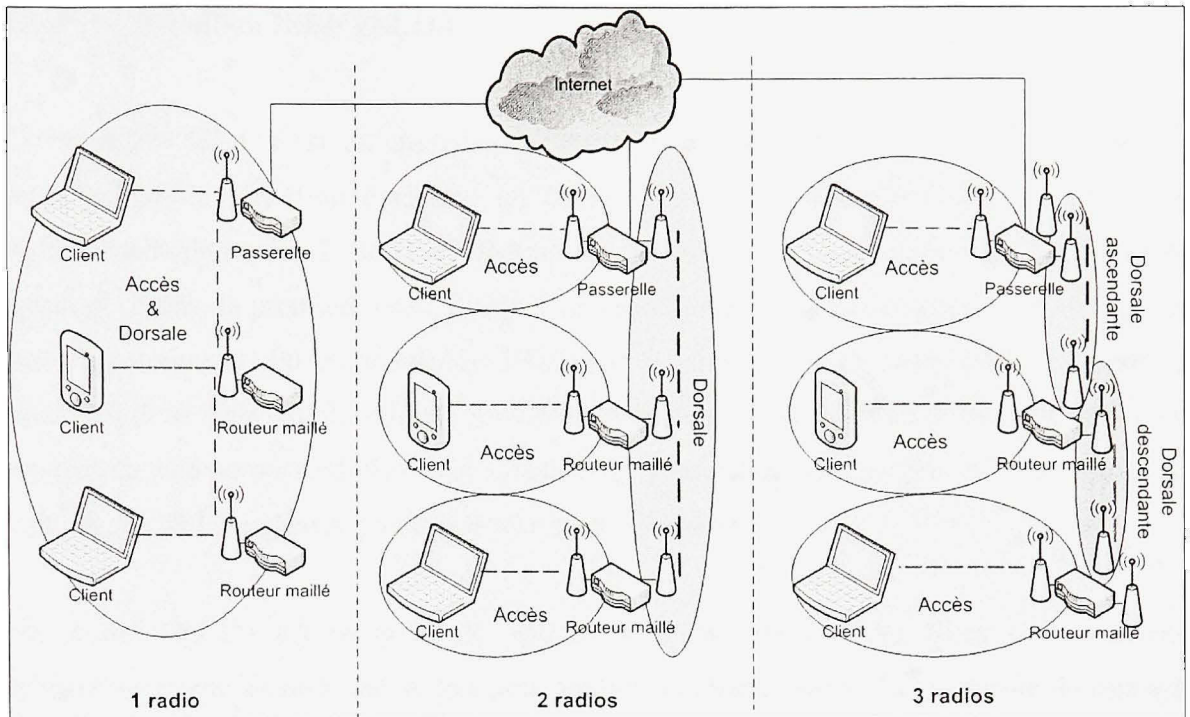


Figure 1.3 Structures de réseau maillé.

1.1.3 Routage

Un réseau maillé est un réseau de routeurs. Les fonctions de routage sont les mêmes que celles rencontrées dans un réseau filaire classique auxquels s'ajoutent des problématiques de performance dues au média radio. L'utilisation de la ressource est optimisée pour ne pas saturer le réseau.

Deux types de routage sont possibles (Joseph D. Camp et Edward W. Knightly, 2008) :

- **Routage proactif** : il établit une topologie en arbre de la passerelle vers les routeurs. Il est mis à jour après chaque modification dans la topologie. La métrique utilisée est le nombre de sauts.
- **Routage dynamique** : les routes entre les clients du réseau maillé ne sont pas maintenues. Elles sont mises à jour après chaque demande de connexion.

1.1.4 Brouillon IEEE 802.11s

Le brouillon 802.11s est un amendement de la norme IEEE 802.11 pour standardiser les réseaux maillés. Il définit comment les commutateurs sont interconnectés pour créer un réseau maillé de niveau 2. Son élaboration a commencé en septembre 2003. Après une série de propositions, la première version D0.01 de brouillon a vu le jour en mars 2006. S'en sont suivis la version D1.00 en novembre 2006 et la version D2.00 en mars 2008. La version actuelle D3.00 (juin 2010) toujours provisoire a été approuvée en mars 2009. Ces multiples reports de validation sont dus à la quantité de paramètres mis en jeu et la pression de compagnies indépendantes à valoriser leur propre solution.

Un réseau 802.11s est composé de stations maillées (mesh STA). Elles se connectent automatiquement les unes aux autres pour former un réseau dorsal. Le protocole de routage utilisé est le protocole HWMP (Hybrid Wireless Mesh Protocol) qui combine un routage proactif et dynamique.

Les stations maillées peuvent être associées à des points d'accès WiFi pour créer le réseau d'accès. Une station maillée connectée au réseau filaire prend le rôle de passerelle.

D'autre part, les spécifications IEEE 802.11s envisagent des mécanismes de qualité de service (QoS) et de contrôle de la congestion qui n'ont pas encore été validés. Ils doivent permettre l'optimisation de la répartition de la bande passante entre clients et trafics.

1.1.5 Technologies multiradio

En parallèle avec les efforts de 802.11s de nombreuses solutions non standardisées ont été développées. La solution la plus intéressante est la solution multiradio. Elle se base sur la baisse du coût des technologies radio. Elle crée un réseau maillé de niveau 3 en se basant sur l'utilisation des technologies WiFi validées (IEEE 802.11a/b/g).

L'architecture des réseaux multiradio est de type infrastructure, le réseau dorsal (802.11a) est isolé du réseau d'accès (802.11g). Les stations maillées sont équipées de trois radios : une pour le réseau d'accès et deux pour le réseau dorsal (connexions "ascendante" et "descendante") sur des canaux sans interférence. Dans le réseau dorsal, les liens sans fils utilisés sont des connexions WiFi classiques; une radio est utilisée en mode station pour établir la connexion ascendante et l'autre en mode point d'accès pour accueillir la connexion descendante. C'est ainsi que l'on peut caractériser un réseau maillé multiradio comme étant une mise en cascade de connexions WiFi classiques grâce à des routeurs multiradio.

L'étude de la performance menée dans ce mémoire présente les perspectives d'utilisation des réseaux maillés multiradio. Elle propose la mise en place d'un réseau maillé multiradio sans modification des protocoles WiFi sur un banc de test. L'influence du niveau 3 sur la performance est évaluée.

1.2 Techniques d'évaluation

Les réseaux maillés sont en développement depuis 2003. La grande quantité de paramètres mis en jeu et les possibilités de développement ont passionné une génération entière de chercheurs. Aujourd'hui, la solution la plus prometteuse est le multiradio. Les réseaux maillés multiradio offrent de nouvelles perspectives de performance en partageant le trafic entre accès et relayage.

Cette partie propose un tour d'horizon des avancés et découvertes de la recherche autour des réseaux maillés. Elle présente : les différentes approches d'étude de la performance, les axes explorés par la recherche et les directions envisagées pour le futur.

1.2.1 Approches

Les approches en recherche pour l'étude de la performance de réseaux sont de quatre types :

- analytique,
- simulation,

- banc de test,
- environnement réel.

L'étude analytique est basée sur l'élaboration d'algorithmes permettant de répondre mathématiquement à des problèmes précis. Par exemple, le développement d'un algorithme de routage permettant de déterminer le plus court chemin au travers d'un réseau maillé en se basant sur une métrique de bande passante et délai.

L'étude en simulation est basée sur des tests en environnements simulés (Elias Weingärtner, Hendrik vom Lehn et Klaus Wehrle, 2009). Les comportements sont testés sur une plateforme de simulation telle que NS-2. Ces études permettent une première évaluation de l'apport d'une nouvelle technologie. Elles ont l'avantage d'être simples et rapides de mise en œuvre. Mais les résultats obtenus restent des approximations : les éléments et technologies mis en jeux dans la simulation sont tous des approximations de la réalité.

L'étude sur banc de test est une troisième étape dans le développement d'une nouvelle technologie. Il permet sa mise en place dans des conditions proches de la réalité. Les analyses sont réalisées sur des équipements réels dans des conditions expérimentales. Les trafics et comportements sont simulés par le chercheur. Les résultats obtenus sont très proches des résultats attendus en conditions réelles.

L'étude en environnement réel est l'étape ultime. Elle permet de vérifier effectivement la capacité d'une technologie à répondre aux besoins des utilisateurs. Elle est menée sur un large panel de réseaux en production pour vérifier l'efficacité d'une technologie dans un maximum de contexte possible. Les limites de cette approche sont qu'elle est lourde à mettre en œuvre et difficilement reproductible.

1.2.2 Axes de recherche explorés

Le développement des technologies maillées est le centre de nombreux débats. La norme 802.11s n'étant toujours pas validée, des mécanismes permettant encore et toujours d'améliorer sa performance sont proposés chaque jour. Ils se basent sur la souplesse des paramètres mis en jeux : environnement (interférences, obstacles...), physique (radio, puissance...), liaison (MAC, aiguillage, QoS...), réseau (routage, équité, QoS...) et transport (TCP, UDP...).

Les axes principalement explorés par la recherche sont : l'architecture des réseaux maillés, le partage de la bande passante et les limites de TCP dans un environnement radio.

1.2.2.1 Côté réseau

Le fonctionnement et l'organisation des réseaux maillés sont en constante évolution, tout particulièrement les problématiques de routage. Bicket, Aguayo, Biswas, Sanjit et Morris (2005) élaborent l'utilisation de nouvelles métriques permettant de choisir des chemins en se basant sur la mesure du délai, du nombre de sauts et la répartition de la bande passante entre les clients ce qui implique des échanges d'informations inter couches. Kone, Das, Zhao et Zheng (2007) proposent un protocole de routage QUORUM prenant pour métrique le délai et la robustesse des liens (taux de perte) ceux qui impliquent là encore des échanges entre les couches 1, 2 et 3.

D'autre part, de nouvelles topologies ont vu le jour tel que les réseaux maillés multiradio. L'étude de Ramachandran, Sheriff, Belding et Almeroth (2008) a étudié l'optimisation de la performance sur un réseau de 20 routeurs dans une approche de simulation et a proposé un protocole de gestion de l'attribution des canaux automatique qui a permis de réduire les interférences et d'accroître la performance des liens.

1.2.2.2 Côté gestion de trafic

La répartition de la bande passante est l'un des grands défis de la gestion du trafic. Dans un réseau maillé, un lien internet est partagé entre une multitude de clients connectés sur les différents routeurs d'accès. Le réseau se doit de garantir le partage équitable de la bande passante Internet entre l'ensemble de ses clients; un client éloigné ne doit pas avoir accès à moins de bande passante qu'un client proche du lien Internet.

Les premières études sur le partage de la bande passante portent sur la gestion de l'équité. Badarla, Malone et Leith (2008) testent le protocole 802.11e. Ils ont configuré les trafics TCP ACKs prioritaires et utilisé le protocole TXOP pour gérer la répartition de la charge. Les résultats montrent une amélioration de l'équité. Janghwan et Ikjun (2008) proposent un algorithme de répartition de charge. À chaque routeur du réseau est attribuée une part de la bande passante totale proportionnelle au nombre de ses clients. L'équité est là encore améliorée. Mais ces mécanismes ont un coût sur la performance du réseau. Ils impliquent de nombreux échanges entre les routeurs pour synchroniser leurs politiques.

La gestion de la QoS a aussi été étudiée. Jangeun et Sichitiu (2003) proposent la combinaison de mécanismes de gestion de l'équité au niveau 3 et de gestion de la QoS au niveau 2. Les résultats obtenus sont très probants. Mais ils impliquent des modifications du protocole MAC et de l'infrastructure de routage.

1.2.2.3 Côté trafics (TCP)

TCP est un protocole de transport très sensible à la perte. Tout paquet retardé, réordonné ou perdu impose le renvoi de la fenêtre complète sauf si SACK (Selective Acknowledgment) est utilisé. SACK permet à la réception d'informer uniquement des paquets perdus, mais son utilisation est rare. Li-Ping, Wei-Kuan, Te-Chung, Sun et Meng Chang (2007) présentent une étude en simulation de la dégradation de la performance de TCP au travers d'un réseau maillé multisaut. Les résultats montrent que la performance se dégrade rapidement : après seulement deux sauts, elle est divisée par deux. Koutsonikolas, Dyaberi, Garimella, Fahmy et

Hu (2007) présentent une optimisation des paramètres RTS/CTS et de la taille de la fenêtre TCP. Les résultats démontrent qu'il est possible d'augmenter la performance de TCP en modifiant la taille de sa fenêtre de congestion. Mais toutes ces modifications impliquent des modifications du protocole TCP. Ces modifications sont rendues impossibles aux vues des impacts mondiaux qu'elles représentent.

Ces études ont montré que la dégradation de performance de TCP était principalement due aux problèmes de pertes dans le réseau WiFi. L'environnement extérieur et les interférences sont les principales causes de ses dégradations. Plus il y a de sauts, plus leur probabilité est grande et moins TCP est performant. L'influence de la technologie WiFi sur la performance est donc faible. Pour une même topologie, la dégradation de la performance de TCP est similaire dans les réseaux maillés 802.11s et multiradio.

1.2.3 Objectifs

Les travaux de recherche menés ont montré qu'il est possible d'améliorer la performance des réseaux maillés en modifiant certains paramètres mis en jeu. Néanmoins, ces changements impliquent des modifications dans la structure des protocoles. Ces modifications sont aujourd'hui difficilement envisageables, car elles impliquent un coût trop important sur la structure de réseaux déjà déployés.

Notre étude propose une approche plus pratique dans l'optimisation de la performance des réseaux maillés. Il est proposé de maximiser l'utilisation de l'existant en modifiant uniquement des paramètres de niveau 3. L'étude est menée sur un banc de test mettant en jeu des équipements réseaux réels. Les routeurs utilisés sont des routeurs multiradio équipés de deux radios. Des études critiques de performance sont mené sur des topologies arbre et multisaut se qui nous a permis de valider l'apport de mécanismes de QoS de niveau 3 sur les réseaux maillés.

1.3 Performance : définition et mesures

La performance d'un réseau est une évaluation de sa capacité à répondre aux besoins des clients. Les problèmes sont différents en fonction de l'utilisation d'un réseau. Par exemple, un réseau de téléphonie VoIP doit offrir des délais de bout en bout minimaux pour des communications de qualité.

Les réseaux maillés sont utilisés pour des trafics de données, de téléphonie et de vidéo non direct (streaming). La performance dépend du RTT qui doit être inférieur à 300ms pour les communications VoIP, de la bande passante qui doit être suffisante pour le transfert de vidéos (ex : youtube), de la stabilité du réseau et de l'équité entre les clients.

1.3.1 RTT

Le RTT (Round Trip Time) est le temps d'aller et retour entre un émetteur et un destinataire. Il est critique pour des communications voix (VoIP). Il doit être minimal pour ne pas dégrader le service de l'utilisateur.

On peut mesurer le délai à l'aide du protocole ICMP (Internet Contrôle Message Protocol). Son action correspond à l'envoi d'un message « Echo Request » à un destinataire qui y répond immédiatement par un message « Echo Reply ». Le délai entre l'envoi de la requête et la réception de la réponse est le RTT.

Dans l'étude, le RTT est mesuré « à vide » entre un client et la passerelle de sortie du réseau.

1.3.2 Bande passante

La bande passante est la mesure du débit total offert par le réseau pour rejoindre l'Internet. Elle dépend de la topologie du réseau, de la configuration des liens et de l'environnement extérieur (interférences, obstacles...). La bande passante utile de TCP est environ égale à 80%

de la bande passante réelle. Cet écart est dû à la part de bande passante consommée par les couches inférieures (physique, liaison et réseau).

Pour mesurer la bande passante, le protocole TCP est utilisé pour sa sensibilité aux conditions du réseau. Il s'adapte à l'aide de l'algorithme de gestion de trafic suivant (Larry L. Peterson et Bruce S. Davie, 2000) :

$$\text{Débit} = \frac{W \text{ MSS}}{RTT} \quad (1.1)$$

Le RTT représente le temps d'aller et retour entre l'émission et la destination. W représente la taille de la fenêtre de flux. Il correspond au nombre de paquets envoyés pendant un RTT. Il est adapté en fonction du taux de perte de la connexion. MSS (Maximum Segment Size) correspond à la taille maximale de segment TCP. Elle est généralement fixée à 1492 octets. La performance de TCP est ainsi liée au taux de perte et délais dans le réseau.

Dans l'étude, la bande passante est évaluée « à vide » pour chaque client. Le trafic TCP est mis en place successivement de la passerelle Internet vers les clients.

1.3.3 Stabilité

La stabilité du réseau est une évaluation temporelle de la performance du réseau. Elle consiste à mesurer sur une longue période les écarts de RTT et de bande passante. Plus les écarts sont importants, moins le réseau est stable.

La mesure est réalisée par l'analyse de l'évolution des trafics TCP. Les trafics sont établis pour une durée de 10 minutes entre clients et passerelle.

1.3.4 Équité

L'équité permet d'évaluer la capacité du réseau à partager la bande passante équitablement entre les clients.

L'indice d'équité standard est utilisé (Larry L. Peterson et Bruce S. Davie, 2000) :

$$\frac{[\sum_{i=1}^n (x_i)]^2}{n \sum_{i=1}^n (x_i^2)} \quad (1.2)$$

Les x_i sont les valeurs moyennes des trafics TCP. La valeur de l'indice est comprise entre 0 (pas d'équité) et 1 (équité parfaite).

Dans l'étude, des trafics TCP sont exécutés simultanément entre la passerelle et tous les clients. Après 10 minutes, les moyennes de chaque trafic sont utilisées pour le calcul de l'indice d'équité.

1.4 Banc de test : exemple de MiNT

Le système MiNT (De, Raniwala, Sharma et Chiueh (2005)) a été étudié pour ses fortes ressemblances avec notre problématique d'analyse de performance. MiNT automatise la mise en place d'expériences de réseau maillé sur banc de test dans une approche utilisateur intuitive et graphique. Mais malheureusement, son architecture le limite à l'étude de la performance des niveaux 1 et 2.

MiNT offre ainsi la possibilité d'automatiser des analyses de performance sur un banc de test de petite taille. Le banc de test est un réseau WiFi maillé multisaut à une antenne par nœud. Les effets de la mobilité sont analysés grâce à des robots mobiles sur lesquels sont installées les antennes.

1.4.1 Architecture

MiNT (Figure 1.4) est composé d'un ensemble de nœuds de test (*core node*) contrôlés par un nœud de contrôle (*central controller*). Les nœuds de test forment le banc de test maillé. Ils sont équipés d'interfaces WiFi connectées à des antennes atténuées. Les antennes sont installées sur des robots mobiles pour simuler la mobilité. La deuxième antenne permet d'enregistrer (*sniffing*) le trafic transitant pour les mesures de performance. Le nœud de contrôle communique avec les nœuds de test au travers d'un réseau de contrôle indépendant du banc de test.

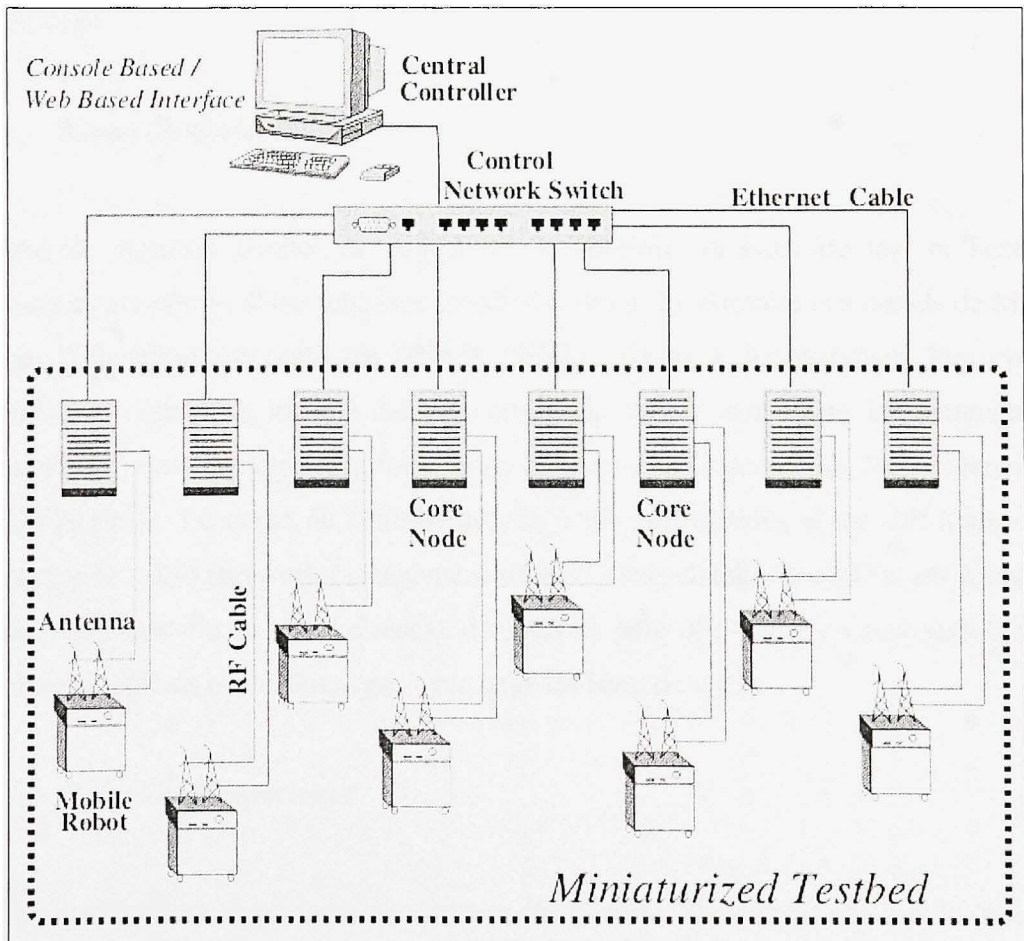


Figure 1.4 Architecture de MiNT.
Tirée de De, Raniwala, Sharma et Chiueh (2005, p. 2)

1.4.1.1 Nœud de test

Un nœud de test est un ordinateur de bureau équipé de deux cartes WiFi NetGear MA311. Pour permettre la mise en place d'un réseau maillé sur une surface réduite, des antennes atténuées sont connectées aux interfaces WiFi. Elles réduisent la puissance du signal de manière telle que le signal ne parcourt pas plus de quelques dizaines de centimètres.

Pour simuler la mobilité, les antennes sont installées sur des robots LEGO (Hobby robots) contrôlés à distance par infrarouge. Les nœuds de test se positionnent automatiquement dans le banc de test.

1.4.1.2 Nœud de contrôle

Le nœud de contrôle permet de centraliser le contrôle du banc de test et l'exécution d'expériences au travers d'une interface terminal et Web. Le contrôle des nœuds de MiNT se base sur l'utilisation du protocole SNMP (Simple Network Management Protocol). Un gestionnaire de nœud est installé dans les nœuds de test. Il met à jour les paramètres des interfaces WiFi en fonction des informations contenues dans les tables MIB (Management Information Base). Le nœud de contrôle accède à ses informations et les met à jour par un gestionnaire de nœud (network management station). Ces échanges transitent sur le réseau de contrôle, isolé du banc de test. Le réseau de contrôle peut être filaire ou non dans la mesure où les canaux utilisés n'interfèrent pas avec ceux du banc de test.

1.4.2 Contrôle d'expérience

La définition d'un test de performance est une succession d'étapes réalisées par le chercheur : configuration de la topologie, des trafics, de la mobilité, des paramètres WiFi et exécution de l'expérience.

1.4.3 Configuration de la topologie

La configuration de la topologie est réalisée manuellement au travers de l'interface graphique GUI (Graphical User Interface). Les nœuds sont représentés par des cercles positionnés à la souris sur le banc de test. Les connexions sont automatiques en fonction du positionnement.

1.4.3.1 Configuration des trafics

Les trafics sont établis par des applications contenues dans la librairie de MiNT. Les sources et puits sont positionnés à la souris sur la topologie.

1.4.3.2 Configuration de la mobilité

La mobilité est configurée par l'édition d'un script. Les durées et trajets sont spécifiés pour chaque nœud.

1.4.3.3 Paramètres WiFi

Les paramètres WiFi sont configurés automatiquement par l'application. Un seul canal est utilisé pour l'ensemble du banc de test. L'utilisateur peut modifier les paramètres (puissance, débit, canal...) en sélectionnant un nœud.

1.4.3.4 Exécution de l'expérience

Une fois la configuration terminée, l'expérience est exécutée. Les traces du trafic sont automatiquement relevées par l'antenne de mesure et centralisées dans le nœud de contrôle.

1.4.4 Traitement des résultats

1.4.4.1 Collection des traces

Au cours de l'expérience, le trafic est enregistré sur la deuxième interface des nœuds en utilisant l'application tcpdump. Tous les paquets reçus au niveau de l'antenne de mesure sont sauvegardés. Une fois l'expérience terminée, ils sont automatiquement centralisés sur le nœud de contrôle.

1.4.4.2 Agrégation des traces

Sur le nœud de contrôle, les traces sont fusionnées en résultats lisibles. Chaque paquet est associé à son parcours temporel dans le réseau. Il est ainsi possible de retracer le trajet précis de tous les paquets dans le réseau.

1.4.4.3 Visualisation des résultats

Les résultats sont visualisés sur l'interface graphique. Ils sont présentés sous une forme comparable au système de simulation NS-2. Le déplacement temporel des paquets est représenté sur la topologie du banc de test. Il est ainsi possible de suivre l'évolution de la performance au niveau du transit des paquets.

1.4.5 Limites et problèmes

Le système MiNT est un système très précis et performant dans l'analyse de la performance d'un réseau maillé multisaut. Par contre, MiNT n'est pas adapté pour des analyses de réseau multiradio. Il ne s'intéresse qu'à des problématiques de niveau 1 et 2. Il ne permet pas la mise en place de mécanisme de routage et de gestion de trafic de niveau 3.

D'autre part, MiNT n'a pas une approche synchronisée. Pour communiquer avec les nœuds, il utilise le protocole SNMP. Les tables MIB sont mises à jour et alors le gestionnaire de nœud peut mettre à jour les paramètres du nœud.

Enfin, le traitement des données n'est pas adapté à des tests de haut niveau. L'analyse paquet est lourde et fastidieuse lorsque les scénarios de trafic se déroulent sur plusieurs minutes.

D'autres systèmes d'expérimentation sur banc de test existent, mais l'étude s'est limitée au système MiNT qui était le système se rapprochant le plus de nos problématiques d'expérimentation.

1.5 Conclusion

Ce chapitre a présenté les technologies de réseaux maillés. Les premières générations utilisaient une radio partagée entre accès et relayage. Puis, des structures à deux et trois radios se sont développées grâce à la baisse du coût des technologies radio. Elles ont permis de séparer l'accès du relayage et même de séparer le relayage ascendant du relayage descendant. De nouvelles limites de performance ont alors vu le jour dû aux interférences engendrées par l'utilisation de plusieurs radios.

En recherche, la plupart des analyses de performance ont été réalisées dans des environnements simulés, simples de mise en œuvre. Ces études ont apporté une première évaluation sur les enjeux du développement de mécanismes de gestion de trafic. Néanmoins, leurs résultats sont limités parce que les simulations sont des approximations de comportements réels. Les études sur banc de test donnent des résultats plus pertinents, mais sont malheureusement encore aujourd'hui difficiles de mise en œuvre. Les solutions d'automatisation telle que MiNT ont été développées pour répondre à des problématiques précises dans l'analyse de la performance.

Ce mémoire propose ainsi d'étudier la performance des réseaux WiFi maillés multiradio sur banc de test. Le banc de test est contrôlé par une plate-forme "maison" ouverte. L'étude met en valeur les limites de performance des technologies actuelles et propose des mécanismes de niveau 3 pour les contrôler. L'étude est basée sur l'utilisation d'équipements et de protocoles existants et validés.

CHAPITRE 2

PLATE-FORME DE CONTRÔLE DE BANC DE TEST

Au fil des années, des solutions pour automatiser les expériences réseau ont vu le jour tel que MiNT. Elles permettent de déployer rapidement des tests dans des conditions proches de la réalité. Mais, elles ont pour la plupart été développées pour répondre à des problématiques particulières. L'étude de MiNT a permis de montrer qu'elle est difficilement transférable à d'autres contextes expérimentaux tels que le nôtre. Or, le besoin croissant des utilisateurs impose à la recherche des délais de développement de plus en plus courts. Développer des applications de gestion de banc de test pour chaque nouvelle problématique devient rébarbatif et non productif.

La plate-forme expérimentale proposée dans ce chapitre est une solution ouverte qui se base sur l'utilisation de l'existant. Les équipements réseau utilisent des systèmes d'exploitation Unix qui ont déjà fait leur preuve. Un large éventail d'outils est disponible pour réaliser tout type de mesure. De plus, les systèmes Unix sont ouverts, ils permettent aux chercheurs d'implanter des protocoles en développement. La plate-forme a ainsi été développée pour synchroniser l'utilisation de ces équipements existants et de leurs outils dans le cadre d'expériences sur banc de test.

La plate-forme permet d'évaluer la performance d'un réseau dans un contexte proche de la réalité. Contrairement à des outils tels que MiNT qui sont spécialisés dans l'analyse des comportements de niveau 1 et 2, la plate-forme s'intéresse aux problèmes de performance des utilisateurs. Les comportements de niveaux 2, 3 et 4 sont analysés.

Ce chapitre présente la plate-forme expérimentale proposée. La présentation commence par les problématiques liées à la mise en place d'expériences sur banc de test. Puis, les propriétés d'une expérience réseau sont définies par l'utilisation d'un script de scénario. Ensuite,

l'architecture et le dessin logiciel de la plate-forme sont présentés. Enfin, une discussion et une conclusion sont proposées.

2.1 Problématique

Les analyses de performance sur banc de test permettent d'évaluer la validité de résultats obtenus par simulation dans un environnement proche des conditions réelles. Les analyses en simulation sont basées sur des approximations de comportements réels. Un environnement réel ayant une complexité quasi infinie, l'étape d'analyse de la performance sur banc de test est essentielle avant la validation d'un protocole ou d'une nouvelle technologie afin de contrôler le plus large nombre de paramètres. Les problématiques liées à ces analyses sont présentées dans cette partie.

2.1.1 Test de performance

Un test de performance réseau sur banc de test est une étude de simulation des conditions réelles. Elle est basée sur l'utilisation d'équipements réels dans des conditions expérimentales. Les éléments et activités sont contrôlés pour la réalisation de tests exhaustifs de limites de performance.

Un test de performance est caractérisé par quatre problématiques : la mise en place du réseau, l'ajout d'éléments de simulation de comportements réseau, le trafic d'exploitation et la mesure de la performance.

2.1.2 Réseau

Le réseau correspond à la mise en place et la configuration du matériel sur le banc de test. Les nœuds sont positionnés et configurés en fonction de la topologie du test.

2.1.2.1 Éléments de simulation

Les éléments de simulation sont des services permettant de simuler des effets particuliers de réseau. Par exemple, un service de taux de perte peut être ajouté à une interface WiFi pour simuler une dégradation de signal. Les éléments de simulation permettent ainsi de tester les effets de comportements réels habituellement non contrôlables de façon contrôlée sur le banc de test.

2.1.2.2 Trafics

Les trafics représentent l'activité du banc de test. Ils sont établis de manière à tester des conditions limites d'utilisation. Par exemple, une grande quantité de flux de trafic utilisateur peut être exécutée simultanément pour évaluer la gestion de l'équité dans le réseau.

2.1.2.3 Mesure

La mesure permet de relever la performance du banc de test qui dépend de la topologie, des éléments de simulation et des trafics. Tout type de mesure peut être envisagé : bande passante, délai, équité... Les outils de mesure sont installés sur les nœuds critiques de la topologie testée. Par exemple, la mesure de la performance d'une connexion TCP est mise en place sur le nœud source du trafic.

2.1.2.4 Banc de test

L'environnement expérimental est composé d'une collection de nœuds organisés dans la topologie de test. Chaque nœud offre des services de configuration d'interfaces, simulation, exécution de trafic et mesure.

2.1.2.5 Nœuds

Les nœuds sont les éléments de base du banc de test. Les services qu'ils offrent permettent l'exécution d'expériences. Il existe trois types de nœud :

- équipement Unix,
- équipement de simulation,
- service extérieur.

Les équipements Unix sont des cartes embarquées ou ordinateurs de bureau équipés d'interfaces réseau. Ils sont pilotés par des systèmes d'exploitation ouverts tels que Linux ou FreeBSD. Ils sont contrôlés localement par leur interface série et à distance par une connexion terminale SSH.

Les équipements propriétaires sont des équipements de simulation réseau. Par exemple, un équipement permettant de simuler les effets d'un réseau WAN entre un port d'entrée et un port de sortie. Les équipements propriétaires sont équipés de systèmes d'exploitation propriétaires à accès limité. Ils sont contrôlés à distance par des protocoles terminaux tels que TCL.

Les services extérieurs sont des équipements non contrôlés par l'environnement expérimental. Ils offrent des services non modifiables. Par exemple un serveur SIP qui offre des comptes de téléphonie permettant d'établir des connexions VoIP entre les nœuds du banc de test.

2.1.2.6 Services

Les services sont des outils exécutés localement sur les nœuds. Ils sont propres à chaque nœud. Ils permettent de réaliser des actions expérimentales. Les services sont répartis en trois classes :

- réseau,

- simulation,
- trafic et mesure.

La classe réseau contient l'ensemble des services permettant la mise en place du réseau. Ils permettent la configuration des interfaces réseau, la configuration du réseau et la définition des chemins dans les tables de routage.

La classe simulation correspond aux services qui simulent des comportements particuliers de réseau. Par exemple, un service de taux de perte qui permet de simuler des effets d'interférence sur une connexion WiFi.

La classe trafic et mesure correspond aux services permettant de créer des trafics et de mesurer la performance. Les trafics sont créés par l'ajout de sources et de puits dans le réseau. La mesure peut être associée soit au service de trafic soit à un élément indépendant. Par exemple, un service d'aspiration de trafic (« sniffing ») qui enregistre tous les paquets transitant sur une interface.

2.1.3 Approche

La plate-forme est développée dans une approche ouverte, portable et intuitive. La structure est simple, elle permet de synchroniser l'utilisation de nœuds et services existants. Le contrôle et la mise en place des expériences sont centralisés dans une interface graphique Web intuitive.

2.1.3.1 Gestion des nœuds

La gestion des nœuds est gérée uniformément par la plate-forme. Le fonctionnement des trois types de nœuds (équipement Unix, équipement de simulation, et service extérieur) est harmonisé. Tous les nœuds se connectent à la plate-forme, donnent la liste de leurs services (interfaces et outils) et attendent des ordres d'exécution du serveur central. Les nœuds sont

comparables aux pièces d'un puzzle. Chaque pièce a une forme particulière et un motif particulier. L'utilisateur prend ces pièces et réalise son propre puzzle (expérience) en associant les formes (service de réseau) et les images (services de simulation, trafic et mesure).

2.1.3.2 Gestion des services

Les services sont les outils logiciels offerts par les nœuds pour la réalisation d'expérience. Ils sont activés par des commandes exécutées localement sur les nœuds.

Deux types de commandes sont déterminés :

- synchrone,
- asynchrone.

Les commandes synchrones ont un temps d'exécution court. Elles sont exécutées puis leurs sorties standard stdout et stderr sont retournées. La sortie standard stdout est la sortie normale, qui contient le résultat de l'exécution. La sortie standard stderr est la sortie erreur, qui est retournée uniquement en cas d'erreur et contient le message associé à l'erreur. Il est possible qu'une commande synchrone exécute un service dont l'action se prolonge dans le temps. Dans ce cas, une deuxième commande synchrone est exécutée pour arrêter le service. Par exemple, pour monter une interface réseau la commande synchrone : « ifconfig eth1 192.168.1.1 up » est utilisée, puis, pour la démonter la commande : « ifconfig eth1 down ».

Les commandes asynchrones ont un temps d'exécution long. Elles sont exécutées en tâche de fond sur le nœud. Deux fins d'exécution sont alors possibles. Dans le cas où elle se termine d'elle-même, elle est attendue par une commande « wait », spécifique à la plate-forme. Par exemple : Le service « iperf -c 192.168.1.1 -t 60 » se termine de lui-même au bout de 60 secondes. D'autre part, une commande qui n'a pas de fin d'exécution (ex : « iperf -s ») est arrêtée par une commande « kill », spécifique à la plate-forme. Les sorties standard stdout et stderr sont retournées à la fin de l'exécution des commandes asynchrones.

2.1.3.3 Ouvert sur l'existant

L'approche ouverte de la plate-forme permet aux utilisateurs de suivre les évolutions des équipements et protocoles en développement. La plate-forme est basée sur l'utilisation de l'existant et permet leur synchronisation dans une approche expérience.

Le choix des nœuds Unix ouverts offre des perspectives intéressantes. Des développements spécifiques peuvent être réalisés et associés à de nouveaux services dans la plate-forme. Les protocoles en développement dont la performance a été validée dans des expériences de simulation peuvent ainsi être mis en place rapidement dans des expériences sur banc de test. La validation de la performance d'un nouveau protocole est facilitée.

La plate-forme ouverte et portable offre de bonnes perspectives d'avenir. Elle se concentre sur la synchronisation des outils existants. L'ajout de nouveaux nœuds et services dans la plate-forme est simple et rapide.

2.2 Expérience

Une analyse de la performance sur banc de test est une expérience réseau, donc multi-nœud. Une expérience est composée de deux étapes, l'exécution et le traitement des données. L'exécution correspond à la mise en place du banc de test. Elle est réalisée par l'exécution d'un script d'expérience. Le traitement des données consiste en l'acquisition des traces, la synthèse des données et la visualisation de la performance.

2.2.1 Exécution

L'exécution d'une expérience est une succession de lignes de commandes organisées dans un script. Chaque ligne de commande est associée à un nœud, un type d'exécution (synchrone ou asynchrone) et pour les commandes asynchrones un identifiant script.

Pour une meilleure compréhension, la présentation du script est accompagnée d'un exemple. L'expérience porte sur l'étude de la performance d'une connexion Ethernet de mauvaise qualité. Les nœuds 1 et 2 sont inter connecté par leurs interfaces Ethernet. Le nœud 2 est supposé à problème, il engendre un taux de perte de 10%. La mesure de performance porte sur un trafic TCP ascendant (nœud 1 vers nœud 2).

Les problématiques du script sont présentées dans l'ordre de leur exécution. Des variables sont d'abord utilisées pour simplifier la configuration et la lisibilité. Les étapes d'exécution du banc de test sont : la mise en place du réseau, le positionnement des outils de simulation et l'exécution des trafics et mesures. Pour finir, des mécanismes additionnels de blocs et boucles sont présentés.

2.2.1.1 Variables

Les variables sont utilisées pour faciliter la lisibilité et la configuration du script. Elles sont déclarées en début de script. Par exemple, pour le test de performance elles sont définies par :

```
Noeud1=00:0d:b9:15:bc:70
Noeud2=00:0d:b9:15:ba:e4
LostRate=0.10
```

Les identifiants des nœuds sont associés aux noms « Noeud1 » et « Noeud2 ». Le taux de perte est configuré à 10%. Ces variables sont appelées dans le script avec la syntaxe : $\$\{\text{« nom de la variable »}\}$.

2.2.1.2 Réseau

La mise en place du réseau est réalisée par la configuration des interfaces et routes. Par exemple pour l'analyse de la performance, le réseau est configuré par :

```

${Noeud1} , sync , ifconfig eth1 192.168.1.1 up
${Noeud2} , sync , ifconfig eth1 192.168.1.2 up

```

Les interfaces eth1 des nœuds 1 et 2 sont configurées sur le réseau 192.168.1.0/24.

2.2.1.3 Outils de simulation

Les outils de simulation sont ajoutés sur le réseau pour créer des effets particuliers sur le réseau. Pour l'exemple, les outils de simulation sont définis par :

```

${Noeud2} , sync , ipfw eth1 plr ${LostRate} src-ip any
dst-ip any

```

Un taux de perte est ajouté à l'interface eth1 du nœud 2. La valeur de la perte est associée à la variable "LostRate" dont la valeur est définie en variable à 10%.

2.2.1.4 Mesure et trafics

Les trafics et outils de mesure sont maintenant mis en place. Ils représentent le scénario d'exécution de l'expérience. Pour l'exemple, la mesure est définie par :

```

${Noeud2} , async , iperf -s , id1
${Noeud1} , async , iperf -c 192.168.1.2 -t 600 -i 10 , id2
${Noeud1} , wait , id2
${Noeud2} , kill , id1

```

Un puits (*sink*) TCP est installé sur le nœud 2. Une source de trafic TCP d'une durée de 600 secondes est mise en place sur le nœud 1 vers le puits. Les commandes utilisées sont des commandes asynchrones. Des identifiants sont utilisés pour gérer la fin de leur exécution. La source (id2) qui a une durée d'exécution déterminée est attendue par un « wait » (ligne 3). Le puits (id1) est détruit à la fin de l'exécution du trafic par un « kill » (ligne 4).

2.2.1.5 Relâchement des nœuds

Maintenant que l'exécution des outils de mesures et trafics est terminée, le banc de test est libéré; les services de simulation et de réseau sont arrêtés successivement par :

```
{Noeud2} , sync , ipfw eth1 destroy
{Noeud1} , sync , ifconfig eth1 down
{Noeud2} , sync , ifconfig eth1 down
```

En premier, le service de simulation "ipfw" est dé-configuré sur le noeud 2. Puis les interfaces sont démontées sur les nœuds : 1 et 2. Les nœuds se retrouvent dans leur état initial. Ils sont prêts à exécuter de nouvelles expériences.

2.2.1.6 Autres

Par ailleurs, nous définissons des mécanismes de blocs et boucles sont définis. Ils permettent de multiplier l'exécution d'un phénomène en changeant une valeur.

Les blocs permettent de mettre une étiquette sur un groupe de commande. Cette étiquette est ensuite utilisée dans les résultats à fin d'identification. Un bloc de résultat est associé à un phénomène, par exemple :


```

Ascendant ::
{
    ${Noeud1} , async , iperf -c 192.168.1.2 -t 600 -i 10
        , id2
    ${Noeud1} , wait , id2
}

```

La mise en place du trafic est étiquetée comme étant « Ascendant ». Un deuxième bloc de trafic descendant pourrait alors être défini et étiqueté « Descendant ».

Les boucles sont des blocs qui sont répétés plusieurs fois en changeant la valeur d'une variable. Les commandes sont étiquetées par le nom et la valeur de la variable courante. Par exemple :

```

FOR lostRate IN [0,0.05,0.10]
{
    ${Noeud2} , sync , ipfw eth1 plr ${lostRate} src-ip any
        dst-ip any
    #...(exécution des trafics)...
    ${Noeud2} , sync , ipfw eth1 destroy
}

```

L'exécution des trafics est répétée trois fois pour des taux de perte de 0%, 5% et 10%. Les résultats sont étiquetés par : « lostRate=0 », « lostRate=0.05 » et « lostRate=0.10 ».

Cette partie a permis de présenter le script qui permet l'exécution de l'expérience. Le script complet de l'exemple du test de la performance d'une connexion Ethernet est présenté dans les annexes. Les blocs et boucles sont utilisés pour tester la performance dans les trois cas de taux de perte : 0%, 5% et 10%.

2.2.2 Traitement des données

Le traitement des données de l'expérience permet l'analyse de la performance. Une partie des données est acquise pendant l'exécution de l'expérience : le journal qui enregistre l'état d'exécution des commandes et les traces qui sont des copies des sorties standard. Les données et résultats sont extraits des fichiers traces à la fin de l'exécution de l'expérience.

2.2.2.1 Journal

Le journal est un fichier tableau au format CSV (les valeurs sont séparées par des virgules) contenant les traces d'exécution des commandes. Il permet une validation des étapes de l'expérience. Chaque commande est associée à : identifiant d'exécution, identifiant de nœud, type d'exécution, emplacement des traces, état de l'exécution et message d'alerte (Figure 2.1).

```
Directory , Command Id , Node Id , Executed type , Command , Status , Warning message
...
~/traces/ , 007-LostRate=0_Ascendant-iperf , 00:0d:b9:15:bc:70 , async , iperf -s , Ok ,
...
```

Figure 2.1 Traitement des données : exemple fichier journal.

L'exécution de la commande « iperf -s » s'est déroulée correctement (« Ok »); il n'y a pas de message d'erreur. L'identifiant est « 007-LostRate=0_Ascendant-iperf » et les traces se trouvent dans le dossier « ~/traces/ ».

2.2.2.2 Traces

Les traces sont des fichiers textes contenant les sorties standard stdout et stderr. Ils sont nommés par l'identifiant de la commande, le nom du bloc et le nom du service. Les extensions « .stdout » ou « .stderr » précisent à quelle sortie standard la trace correspond.

Par exemple, la trace de « 007-LostRate=0_Ascendant-iperf.stdout » correspond à la Figure 2.2.

```

/usr/local/bin/iperf -s
-----
Server listening on TCP port 5001
TCP window size: 64.0 KByte (default)
-----
[  5] local 192.168.1.1 port 5001 connected with 192.168.1.2 port 50390
[ ID] Interval          Transfer      Bandwidth
[  5]  0.0-600.0 sec    3.07 GBytes   51.2 Mbits/sec

```

Figure 2.2 **Traitement des données : exemple fichier trace.**

La première ligne rappelle la commande exécutée. Le reste du fichier est la sortie standard d'iperf.

2.2.2.3 Données

Les données sont des fichiers CSV contenant les informations extraites des traces. Un fichier unique est déclaré par nœud et par service de mesure. Les colonnes correspondent aux différents blocs et boucles exécutés. Leur nom est indiqué sur la première ligne. Par exemple, l'extraction du débit moyen des traces « iperf -s » du nœud 2 donne le fichier présenté en Figure 2.3.

```

LostRate=0_Ascendant , LostRate=0.05_Ascendant , LostRate=0.10_Ascendant
51.2 , 48.0 , 45.2

```

Figure 2.3 **Traitement des données : exemple fichier données.**

Trois colonnes sont définies pour les trois valeurs de taux de perte testées. Les valeurs sont indiquées dans la deuxième ligne.

2.2.2.4 Résultats

Les données brutes sont mises en forme dans des graphiques ou tableaux. Les comportements analysés sont ainsi mis en valeur. Les conclusions sont visuelles. Par exemple, le résultat de l'évolution du débit moyen est maintenant présenté (Figure 2.4).

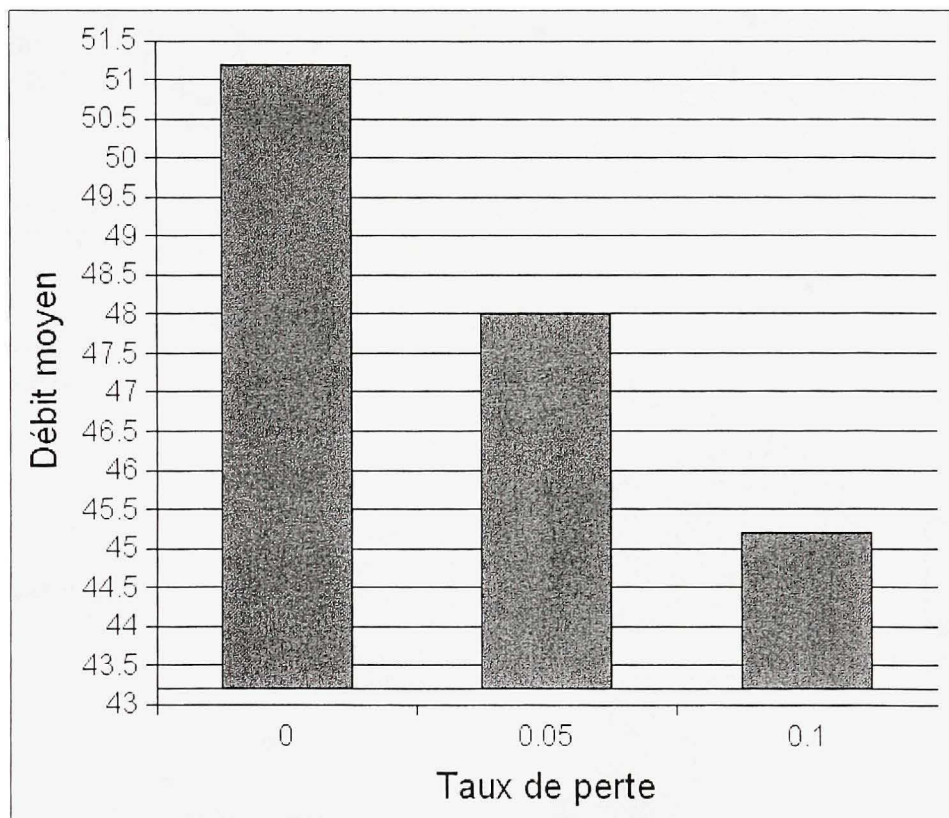


Figure 2.4 Traitement des données : débit moyen VS taux de perte.

Les données sont présentées sous forme de graphique à barres qui permet une appréciation rapide des résultats : le débit diminue avec l'augmentation du taux de perte. La diminution est linéaire et proportionnelle au taux de perte.

2.3 Architecture

L'architecture de l'environnement expérimental est présentée en Figure 2.5. La structure est basée sur une approche équivalente au système d'expérimentation MiNT présenté dans l'étude de l'art. Il est composé de quatre éléments : le banc de test, le réseau de contrôle, le serveur central et l'utilisateur.

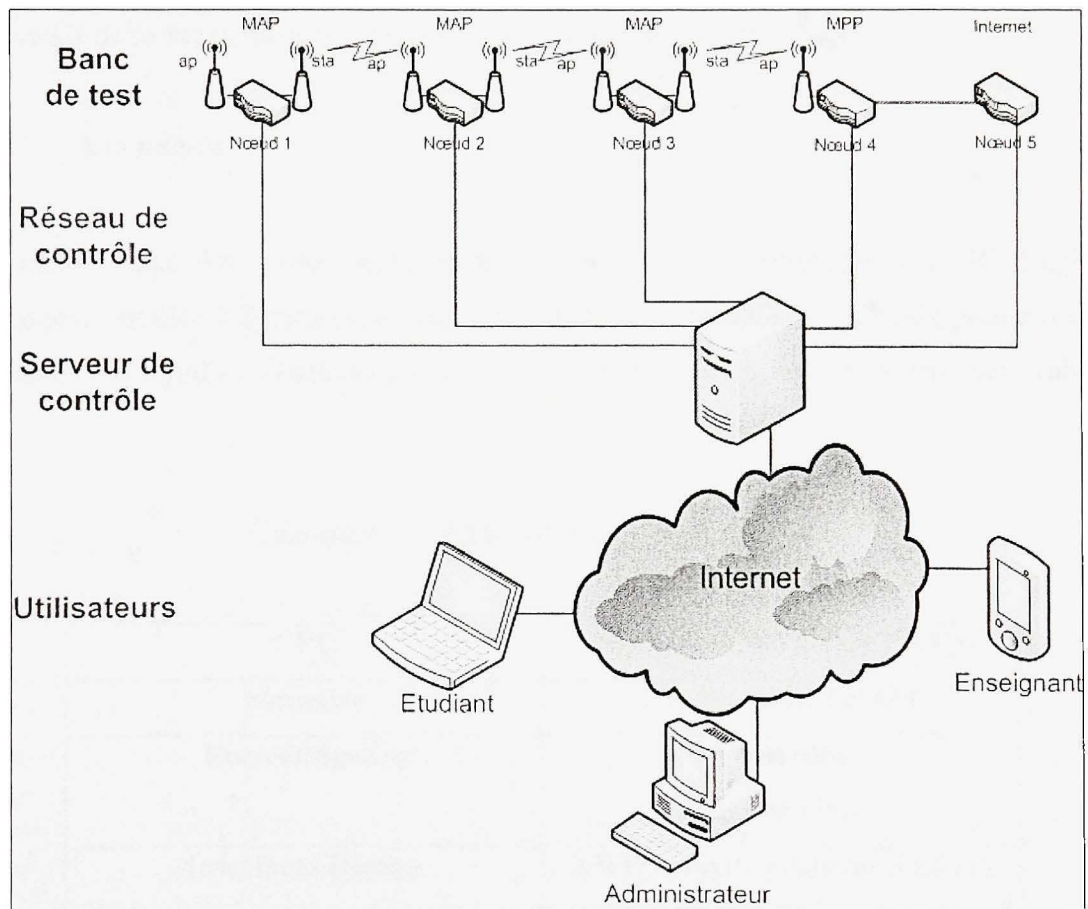


Figure 2.5 Architecture du réseau de contrôle.

Les nœuds du banc de test sont connectés par une interface dédiée au réseau de contrôle. Il est possible d'utiliser une connexion Ethernet ou WiFi dans la mesure où les canaux utilisés n'interfèrent pas avec ceux du banc de test.

Le serveur central permet le contrôle de l'ensemble des éléments. Il fait la transition entre le banc de test et l'utilisateur. Il gère les nœuds connectés au système, les expériences en cours, la définition d'expérience, le traitement des traces et les accès utilisateur.

Les utilisateurs accèdent à l'application à distance au travers d'un navigateur Internet traditionnel (Firefox, Chrome, Internet Explorer...).

Les nœuds de test et le serveur central sont maintenant présentés.

2.3.1 Les nœuds

Les nœuds sont des cartes embarquées développées et distribuées par PC Engines, compagnie installée à Zurich en Suisse. Deux modèles sont utilisés : Alix3d2 pour ses deux emplacements miniPCI (Tableau 2.1) et Alix2d3 pour ses trois interfaces Ethernet (Tableau 2.2).

Tableau 2.1 Caractéristiques carte Alix 3d2

CPU	500 MHz AMD Geode LX800
Mémoire	256 Mo DDR DRAM
Entrées/Sorties	1 port série 2 ports USB
Interfaces Réseau	2 WiFi miniPCI Atheros AR5413 1 port Ethernet
Stockage	1 CompactFlash 1 Go 1 clés USB 256 Mo
Système d'exploitation	FreeBSD 8.0

Tableau 2.2 Caractéristiques carte Alix 2d3

CPU	500 MHz AMD Geode LX800
Mémoire	256 Mo DDR DRAM
Entrées/Sorties	1 port série 2 ports USB
Interfaces Réseau	1 WiFi miniPCI Atheros AR5413 3 ports Ethernet
Stockage	1 CompactFlash 1 Go 1 clés USB 256 Mo
Système d'exploitation	FreeBSD 8.0

Des cartes WiFi Atheros AR5413 (les spécifications sont disponible en annexe) sont installées. Atheros a été choisi pour ses drivers ouverts qui permettent l'implémentation de protocoles en développement.

La clé USB de 256 Mo est utilisée pour supporter l'application maison et les données expérimentales. Elle est isolée du système d'exploitation pour conserver son intégrité.

Le système d'exploitation utilisé est FreeBSD 8.0, il est installé sur le compact flash de 1 Go. Un système Linux (Voyage), plus classique en recherche, avait été envisagé, mais il a montré des limites : blocages et problèmes de stabilité dus au driver MadWiFi en mode point d'accès. Le système d'exploitation FreeBSD 8.0 a été retenu pour sa stabilité et sa performance dans les systèmes embarqués. Le driver Atheros : ath_hal offre un support complet des cartes AR5413 ainsi que la possibilité d'utiliser des protocoles émergents tels que l'IEEE 802.11s.

2.3.2 Le serveur de contrôle

Le serveur central est un serveur Dell, ses caractéristiques sont présentées en Tableau 2.3. La plate-forme est installée sur une machine virtuelle java pour conserver sa portabilité.

Tableau 2.3 Caractéristiques du serveur Dell

CPU	AMD Athlon 64 X2 3800+
Mémoire	2 Go
Interfaces Réseau	3 ports Ethernet
Stockage	1 disque dur 250 Go
Système d'exploitation	Open SuSE 11.0

2.4 Dessin logiciel

Les éléments et mécanismes les plus pertinents de l'application sont présentés dans cette partie. L'application est organisée en quatre groupes distincts (Figure 2.6) : le gestionnaire de nœud, le système de contrôle, la base de données et l'accès utilisateur.

Le gestionnaire de nœud permet l'interaction entre la carte embarquée et le système de contrôle. Il est installé dans chacun des nœuds. Il informe le système des capacités du nœud et attend les commandes à exécuter.

Le système de contrôle est le cœur de la plate-forme. Il gère les interactions entre les expériences et les nœuds. Il permet la mise en place des bancs de test, des outils de mesure, des trafics et l'acquisition des données.

La base de données contient les informations nœud, expérience et administration.

L'accès aux utilisateurs offre un environnement intuitif pour la définition et le contrôle des expériences réseau.

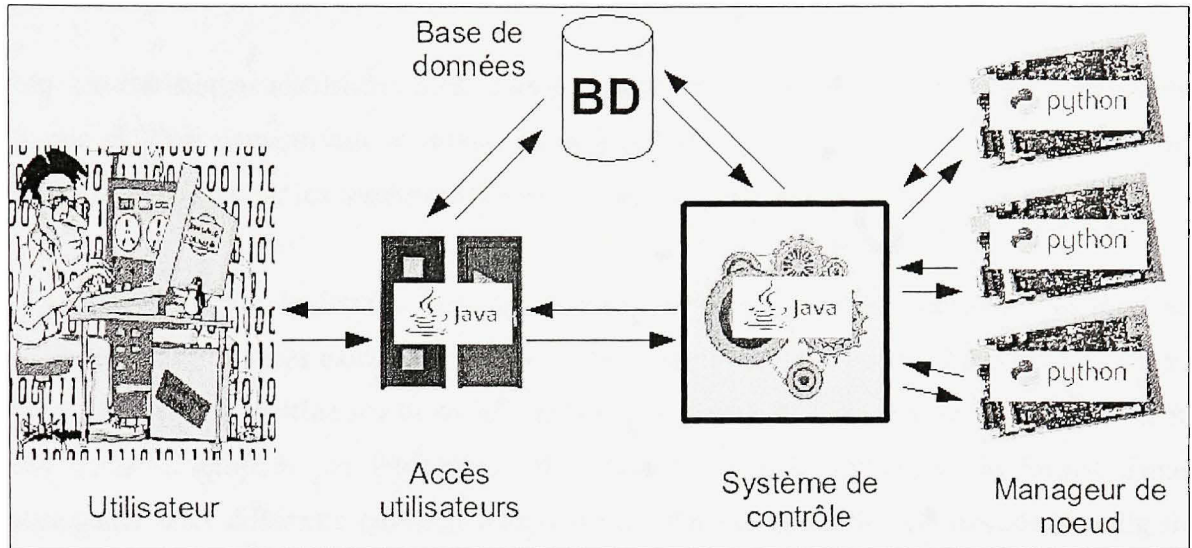


Figure 2.6 Schéma du logiciel.

2.4.1 Le gestionnaire de nœud

Le gestionnaire de nœud est installé sur la clé USB des cartes embarquées. Il est automatiquement exécuté au démarrage après l'initialisation du nœud. Il se connecte au système de contrôle puis attend des commandes à exécuter. Ces mécanismes nécessitent un langage de haut niveau simple et léger. Python a été choisi pour sa syntaxe script, son faible poids d'exécution et sa compatibilité avec de nombreux systèmes d'exploitation.

Le gestionnaire de nœud a trois rôles : établir la connexion avec le serveur, l'informer de ses capacités et exécuter des commandes.

2.4.1.1 Caractéristiques d'un nœud

Un nœud est caractérisé par ses services : matériels et logiciels.

Les caractéristiques matérielles sont la liste des interfaces disponibles sur le nœud, associées à leur modèle, constructeur et driver. Le serveur interprète ces données par une liste de services supportés par les interfaces (Ethernet, point d'accès, station, 802.11s...).

Les caractéristiques logicielles sont déterminées par deux types de services : existant et spécifique. Les services existants sont les outils présents dans le système d'exploitation Unix. Leur présence est vérifiée par l'envoi d'une liste par le serveur. Les services spécifiques sont des outils développés par l'utilisateur. Ils peuvent servir à uniformiser le format d'une commande dans différents systèmes d'exploitation. Par exemple : une commande `ifconfig.sh` permettant de configurer des interfaces WiFi de la même manière sous Linux et FreeBSD. Ils peuvent aussi servir à développer des outils spécifiques à une problématique de performance (protocole, simulation, outil de mesure...).

2.4.1.2 Connexion avec le serveur

Au démarrage, l'application initialise une connexion avec le serveur central, la procédure des échanges d'informations est décrite dans la Figure 2.7.

Dans la phase de connexion, le nœud commence par initialiser une connexion avec le serveur. Puis le serveur l'interroge sur la liste des services qu'il supporte. Le nœud retourne alors la liste complète de ses services. Le nœud est maintenant connecté et prêt à exécuter des expériences.

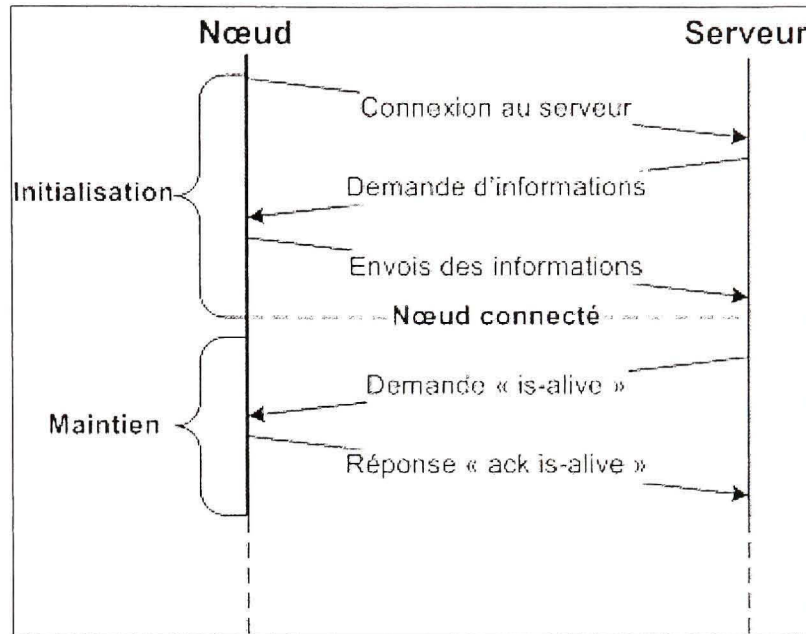


Figure 2.7 Initialisation de la connexion.

C'est alors que la phase de maintien de la connexion commence. Elle est caractérisée par des envois périodiques de messages « is-alive » par le serveur. Le nœud répond avec un message d'acquiescement « ack is-alive ». En cas de 3 non-réponses successives, le nœud est mis en erreur. Après 10, il est déconnecté. Une intervention manuelle est alors nécessaire pour le débloquent.

2.4.1.3 Exécution de commandes

Le nœud est connecté et prêt à exécuter des expériences. Comme il a été vu précédemment une expérience est une succession d'exécutions de commandes. Deux types de commandes sont supportés : synchrone et asynchrone.

Une exécution synchrone est effectuée directement par le gestionnaire de nœud. Il reçoit la commande, l'exécute et retourne les sorties standard au serveur. Pendant l'exécution le nœud

est bloqué : il ne peut recevoir aucune autre commande. Si la commande est trop longue, elle est détruite et un message d'erreur est retourné.

Une exécution asynchrone est sous-traitée à un module exécuté en tâche de fond. Lorsque le gestionnaire reçoit une commande, il lui attribue un identifiant, l'exécute dans le module d'exécution asynchrone et retourne l'identifiant au serveur. Le module exécute la commande, redirige les sorties standard dans des fichiers traces et maintient à jour un statut d'exécution. Une fois la commande terminée le statut est actualisé à « terminé » et le module détruit. Les traces sont transmises au serveur sur demande puis supprimées du nœud.

2.4.2 Le système de contrôle

Pour permettre à la plate-forme d'être utilisée sur dans d'autres systèmes d'exploitation, le système de contrôle a été développé en java, un langage de programmation de haut niveau basé sur une machine virtuelle.

Le système de contrôle est dédié : au contrôle des nœuds, à l'exécution des commandes, au contrôle des expériences, à l'acquisition et extraction des données et à la mise en forme des résultats.

2.4.2.1 Contrôle des nœuds

Le contrôleur de nœud permet le management des nœuds connectés au système. Pour chaque nœud entrant, un module (thread) est créé et exécuté en tâche de fond. Il gère d'abord l'activation du nœud dans le système par l'acquisition de ses services. Ensuite, il vérifie périodiquement l'état de la connexion en envoyant des messages « is-alive ». Il offre au système de contrôle une méthode permettant d'envoyer des messages au nœud et de retourner sa réponse. Ici tout est message : les différenciations entre les types de commandes sont gérées dans le gestionnaire d'exécution.

Les nœuds sont associés à un identifiant et une liste de services. Les interfaces et outils supportés sont connus en permanence par le système.

2.4.2.2 Gestionnaire d'exécution des commandes

Le gestionnaire d'exécution permet de gérer les commandes à exécuter.

Les commandes synchrones sont exécutées comme présentées dans la Figure 2.8. La commande est envoyée au nœud qui l'exécute et retourne immédiatement les sorties standard.

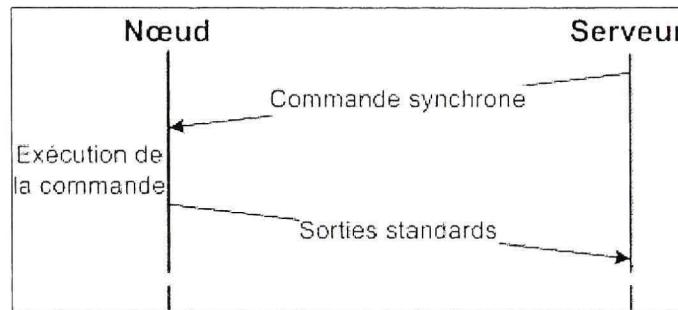


Figure 2.8 Exécution synchrone.

Les commandes asynchrones sont exécutées différemment en fonction qu'elles soient attendues ou détruites.

Les commandes asynchrones avec attente (Figure 2.9) sont envoyées au nœud qui les exécute en tâche de fond. L'identifiant est retourné. La phase d'attente est alors caractérisée par des envois périodiques de demande d'état par le serveur. Quand la commande est terminée, le serveur envoie une requête de sorties standard. Le nœud lui répond et l'exécution de la commande est terminée.

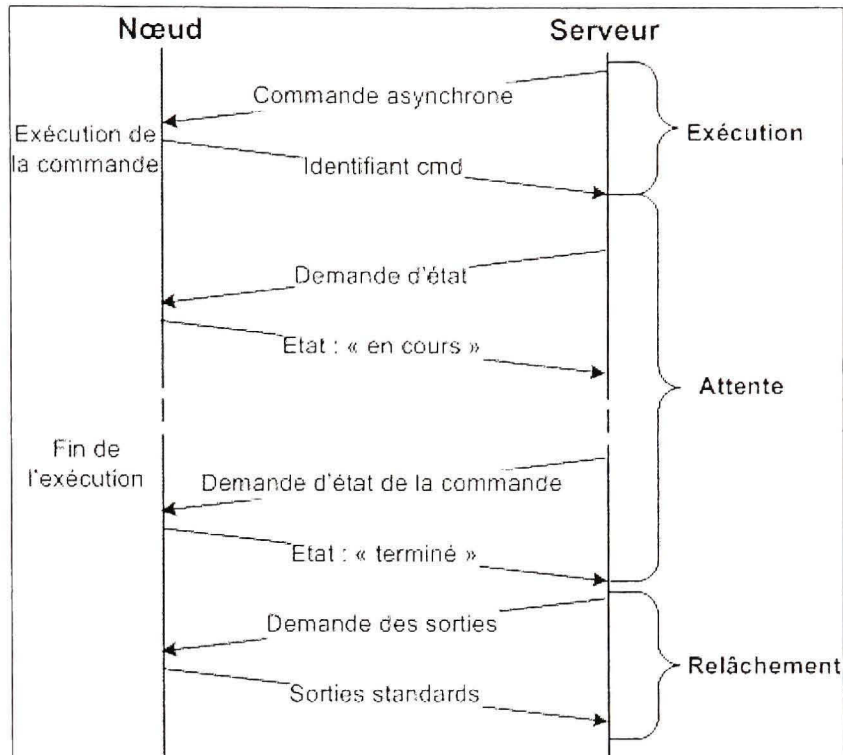


Figure 2.9 Exécution asynchrone avec attente.

Les commandes asynchrones avec destruction (Figure 2.10) sont envoyées au nœud qui les exécute en tâche de fond. L'identifiant est retourné. L'arrêt de la commande est spécifié ultérieurement par l'envoi d'un message « kill » par le serveur. Alors, le nœud arrête la commande et retourne les sorties standard. L'exécution est terminée.

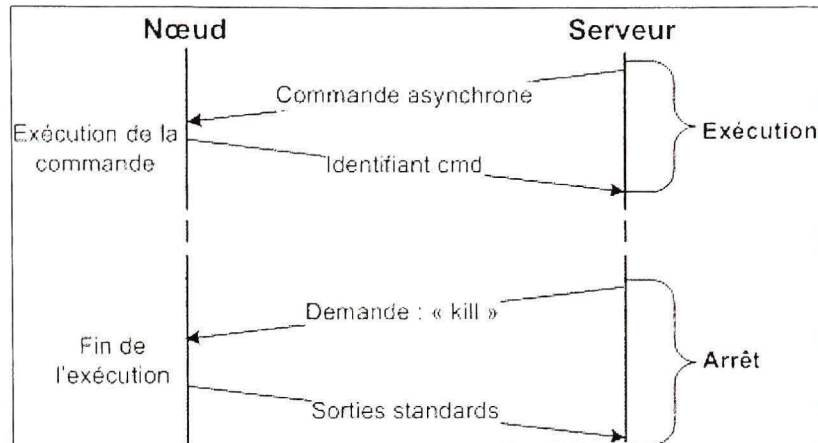


Figure 2.10 Exécution asynchrone avec destruction.

2.4.2.3 Contrôle des expériences

Les expériences sont des scripts. Ils sont exécutés par le serveur indépendamment les uns des autres. Lorsqu'une expérience est démarrée, un module (thread) d'exécution lui est dédié. Il exécute les lignes de commande du script une par une, dans l'ordre de lecture. Le fichier journal est mis à jour au fil de l'exécution du script. Les sorties standard sont sauvegardées dans des fichiers indépendants après chaque fin d'exécution. À la fin de l'expérience, le module est détruit et le statut d'exécution est mis à jour.

2.4.2.4 Acquisition et extraction des données

Les données de l'expérience sont contenues dans les fichiers trace. Après que le script ait été exécuté dans son intégralité et sans erreur, les données sont extraites. Cette action est réalisée par des scripts d'extraction indépendants de la plate-forme. Un script d'extraction est associé à chaque service de mesure.

Les données sont organisées dans des tableaux au format CSV (*Coma-Separated Values*) . Un tableau unique est créé par nœud et par service. Les colonnes représentent les différents

blocs et boucles exécutés lors de l'expérience. Leur identifiant est précisé dans la première ligne de chaque colonne.

Les tableaux sont sauvegardés dans le sous-dossier de l'expérience « data ».

2.4.2.5 Mise en forme des résultats

La dernière étape consiste en la mise en forme des données en résultats lisibles. Pour cela des scripts de résultats indépendants de l'application sont définis par l'utilisateur. Ils sont spécifiques à une configuration d'expérience particulière.

Un script est défini par problématique d'expérience. Les résultats peuvent être présentés sous différentes formes : tableau, graphiques...

Les résultats sont sauvegardés dans le sous-dossier de l'expérience « results ».

2.4.3 La base de données

La base de données fait le lien entre la plate-forme, l'utilisateur et les expériences. Elle est composée de trois parties : nœud, expériences et administration.

La partie nœud contient les informations et services des nœuds associés à leurs identifiants.

La partie expérience contient les données expérimentales. Elles sont organisées en fonction du modèle d'abstraction qui est présenté dans le chapitre suivant. Les informations contenues dans la base de données permettent la génération automatique des scripts d'expérience.

La partie administration conserve les informations de base du fonctionnement de la plate-forme. Notamment les liens entre services et lignes de commandes, les modèles de nœuds supportés et le port du serveur d'accès.

2.4.4 L'accès utilisateur

L'accès des utilisateurs se fait au travers d'une interface graphique/terminal simple et intuitive. Trois types d'utilisateurs sont définis : visiteur, utilisateur et administrateur.

Le visiteur a un accès en lecture seul. Il peut visualiser le contenu des expériences et résultats. Il est par exemple utilisé par le directeur de recherche pour suivre le travail de ses étudiants.

L'utilisateur (chercheur) a un accès en lecture et écriture. Il peut élaborer et exécuter des expériences réseau. Il est utilisé pour mener les tests de performance.

L'administrateur a un accès de contrôle sur la plate-forme. Il peut modifier l'ensemble des paramètres : nœuds, services, commandes... Il est réservé pour un utilisateur aguerri pour des mises à niveau de la plate-forme.

La plate-forme est équipée de deux interfaces d'accès : terminal et Web.

2.4.4.1 Terminal

Le terminal offre un accès rudimentaire à la plate-forme. Il permet de visualiser les nœuds connectés, d'exécuter des expériences et de configurer les paramètres de la plate-forme.

La création et l'élaboration d'expériences sont réalisées soit par une édition manuelle des scripts, soit au travers de l'interface Web.

2.4.4.2 Web

L'interface Web est une interface graphique qui offre un accès intuitif aux différentes options de la plate-forme. Tous les éléments y sont présents : nœuds connectés, expériences et administration.

L'élaboration d'expériences réseau est graphique. L'utilisateur positionne à la souris les éléments du banc de test. L'organisation de l'élaboration suit le modèle d'abstraction présenté dans le chapitre suivant.

Une fois les expériences terminées, l'utilisateur a accès aux données et résultats. En quelques clics il peut visualiser ou télécharger les graphiques générés automatiquement par la plate-forme.

2.5 Discussion

Ce chapitre a présenté une nouvelle approche dans l'exécution d'expériences sur banc de test. La plate-forme proposée est innovante par son architecture modulaire et ouverte à l'existant. Le cœur de la plate-forme se concentre sur la synchronisation des équipements et services ouverts existants par une approche simple : le nœud se connecte au serveur central et attend des services à exécuter. La modularité de la plate-forme se traduit par l'utilisation de scripts extérieurs à la plate-forme utilisés par exemple pour le traitement des données en résultats. Ces scripts sont définis par l'utilisateur en fonction de ses besoins.

D'autre part, la plate-forme garantit la fiabilité des résultats. Toutes les traces d'exécution des commandes sont automatiquement sauvegardées et centralisées. Un journal de bord est mis à jour après l'exécution de chaque commande. L'utilisateur est ainsi en mesure de valider les étapes de son expérience.

La plate-forme permet ainsi au chercheur de consacrer plus de crédit à sa problématique de recherche et moins aux techniques de mise œuvre du banc de test. La grande souplesse de la plate-forme lui permet d'ajouter à sa guise de nouveaux équipements et services. La modularité lui permet de définir précisément les résultats qu'il attend. La plate-forme s'adapte aux problèmes d'aujourd'hui et de demain.

2.6 Conclusion

Dans ce chapitre, la plate-forme expérimentale a été présentée. Elle permet d'automatiser la mise en place d'expériences réseau en utilisant des équipements existants. Les nœuds sont configurés à partir des informations contenues dans un script d'expérience. Les données sont centralisées dans le serveur de contrôle et les résultats automatiquement générés en fonction du besoin de l'utilisateur.

La plate-forme offre une approche utilisateur intuitive. Les mises en place d'expérience ainsi simplifiées permettent au chercheur de se concentrer plus à sa problématique de recherche et moins à la méthodologie des tests.

La plate-forme est portable. Les langages utilisés java et Python sont supportés sur la plupart des systèmes d'exploitation. L'ajout de nouveaux équipements est facilité par une simplicité du gestionnaire de nœud. Son adaptation nécessite des connaissances intermédiaires dans les systèmes Unix.

La plate-forme est aujourd'hui (juin 2010) à sa première version 1.0. Elle est capable d'exécuter des scripts d'expérience, de centraliser les données et de générer des résultats. L'interface utilisée est une interface terminal. L'interface Web graphique et la base de données ne sont pas encore terminées. Leur développement fait partie de futures améliorations qui seront apportées à la plate-forme prochainement. D'autre part, l'utilisation des équipements de simulation et services extérieurs n'a pas encore été testée. Ils feront parties de futurs problèmes expérimentaux.

CHAPITRE 3

PERFORMANCE DES RESEAUX WIFI MAILLES MULTIRADIO

Ce chapitre présente les résultats de l'analyse de la performance des réseaux maillés multiradio. Ces réseaux sont composés de deux sous-réseaux : le réseau dorsal permettant de relayer le trafic entre les clients et l'Internet et le réseau d'accès permettant d'accueillir les clients. Les nœuds multiradio offrent la possibilité d'utiliser une ou plusieurs radios permettant de réduire les interférences en utilisant des canaux dédiés. L'étude de la performance porte sur le réseau dorsal. Les nœuds sont équipés de deux antennes. Une radio est dédiée aux trafics ascendants (vers l'Internet) et l'autre aux trafics descendants (vers les clients).

Ce chapitre est organisé en cinq parties. La première présente le paramétrage des équipements réseau. La deuxième permet une évaluation de la gestion de l'équité dans les réseaux maillés. Les paramètres étudiés sont : les sauts WiFi, l'Internet, le taux de perte et le partage de la bande passante. La troisième présente des mécanismes de QoS permettant d'améliorer la gestion de l'équité. Puis, une discussion est développée sur une implémentation en environnement réel. Enfin, une discussion et une conclusion sont proposées.

3.1 Paramétrage

La première topologie étudiée, composée de six nœuds, est dite multisaut (Figure 4.1). Trois rôles y sont joués : MP (Mesh Point), MPP (Mesh Portal Point) et Internet. Le MP est un routeur de relais dans le réseau dorsal. Il possède deux interfaces WiFi : un AP pour le trafic descendant et un STA pour le trafic ascendant. Le MPP est un routeur de passerelle entre le réseau dorsal (AP WiFi) et le réseau Internet (Ethernet). Le rôle Internet, comme son nom l'indique simule les effets du nuage Internet (délai, bande passante...).

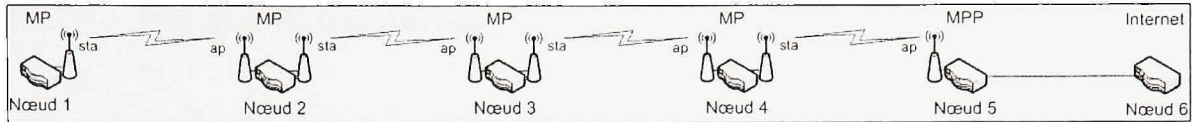


Figure 3.1 Topologie multisaut.

La performance des équipements WiFi est influencée par des paramètres internes et externes. L'environnement de test utilisé est une salle de laboratoire d'une surface de 48m² situé dans les locaux de l'INRS. À l'intérieur de la pièce, aucun autre point d'accès n'était activé. Par contre, il y avait une présence très importante de réseaux étrangers.

Les mesures ont été effectuées sur des trafics TCP. La bande passante est relevée avec iperf à intervalles de 10 secondes sur une durée de 10 minutes (600 secondes). Les 100 premières et dernières secondes sont éliminées éviter les phases transitoires.

Le délai est mesuré avec l'outil ping sur 100 échantillons.

L'équité est mesurée entre les trafics TCP avec l'indice d'équité standard présenté dans la revue de la littérature.

Les paramétrages qui suivent ont pour objectif de réduire l'impact des perturbations extérieures et intérieures du banc de test. Les paramètres qui sont analysés sont : le débit, la puissance, les canaux et paires de canaux sans interférences.

3.1.1 Débit

Les normes WiFi 802.11g et 802.11a proposent des débits de commutation de 1 à 54 Mbps. Dans une configuration classique, le débit est déterminé automatiquement par le pilote (*driver*) en fonction du taux de perte. Les débits plus élevés, plus sujets aux perturbations extérieures sont utilisés dans des conditions « idéales ». Dans la topologie, les nœuds sont peu éloignés les uns des autres. Le débit a été configuré à 54 Mbps.

3.1.2 Puissance de transmission

Des puissances de 0 dbm à 23 dbm sont supportées par les cartes WiFi. Encore une fois, en conditions normales, le pilote détermine automatiquement son niveau en fonction du taux de perte. Dans cette étude, les distances de parcours très faibles permettent de configurer la puissance de transmission à la valeur minimale permise par le pilote : 0 dbm.

3.1.3 Canaux

L'impact de l'environnement extérieur est évalué par une étude de la performance des canaux. Les résultats présentés dans le Tableau 4.1 mettent en valeur des différences de comportement entre le 802.11g et le 802.11a.

Le 802.11g offre des performances variables. La bande passante comprise entre 8,04 Mbps et 24,41 Mbps est très instable avec des variations supérieures à 5%. Le délai mesuré est d'environ 1 ms avec de grandes variations. Ces faibles performances traduisent la concurrence d'un grand nombre de réseau WiFi sur le 802.11g (2.4 GHz).

Le 802.11a offre des performances plus stables. La bande passante est supérieure à 27 Mbps avec des variations faibles, inférieures à 3%. Le délai est largement inférieur à 1 ms. Ces bonnes performances traduisent la faible utilisation du 802.11a (5 GHz).

Ces résultats rappellent la large popularité des technologies du 802.11g face à celles du 802.11a. Pour cette raison, le 802.11a est utilisé dans le réseau dorsal pour ses bonnes performances. Quant au 802.11g il est utilisé dans le réseau d'accès pour sa large popularité dans les équipements mobiles.

Tableau 3.1 Performance des canaux WiFi (Mbps)

		Débit (Mbps)		RTT (ms)	
	Canaux	Moyenne	Écart type	Moyenne	Écart type
802.11g	1	8,04	1,89	2,71	5,02
	2	22,23	1,58	0,84	1,28
	3	24,41	1,25	0,52	0,48
	4	23,16	0,99	0,49	0,48
	5	21,39	1,13	0,70	0,98
	6	13,57	1,38	2,69	4,78
	7	22,76	0,93	0,83	1,18
	8	21,21	2,23	0,75	0,88
	9	22,13	0,98	0,65	0,77
	10	10,08	2,00	0,91	0,74
	11	16,28	1,04	1,81	4,90
802.11a	36	27,32	0,71	0,38	0,25
	40	27,02	0,65	0,37	0,19
	44	27,77	0,72	0,36	0,18
	48	28,11	0,70	0,39	0,31
	52	28,96	0,41	0,37	0,14
	56	28,15	0,76	0,39	0,23
	60	27,69	0,53	0,36	0,18
	64	27,50	0,59	0,39	0,32

3.1.4 Interférences

Nous portons maintenant l'attention sur l'étude de la performance du banc de test. Les réseaux multisaut impliquent l'utilisation de plusieurs canaux simultanément. Les interférences qu'ils produisent entre eux doivent être minimales. Pour l'étude des interférences, la topologie présentée en Figure 4.2 est utilisée. La performance des deux

trafics en parallèle (1-2 et 2-3) est mesurée. Deux scénarios sont étudiés : les interférences entre le réseau dorsal et le réseau d'accès, entre 802.11a et 802.11g (Tableau 4.2) et les interférences au sein du réseau dorsal, inter 802.11a (Tableau 4.3). Dans ces tableaux, les deux valeurs du dessus correspondent au débit moyen des trafics 1 et 2, la troisième du dessous correspond à la moyenne des deux trafics.

L'étude du réseau dorsal VS le réseau d'accès montre une performance stable pour le 802.11a et variable pour le 802.11g. Le comportement est identique à celui étudié au niveau des canaux. Il n'y a donc pas d'interférence entre le 802.11g et le 802.11a.

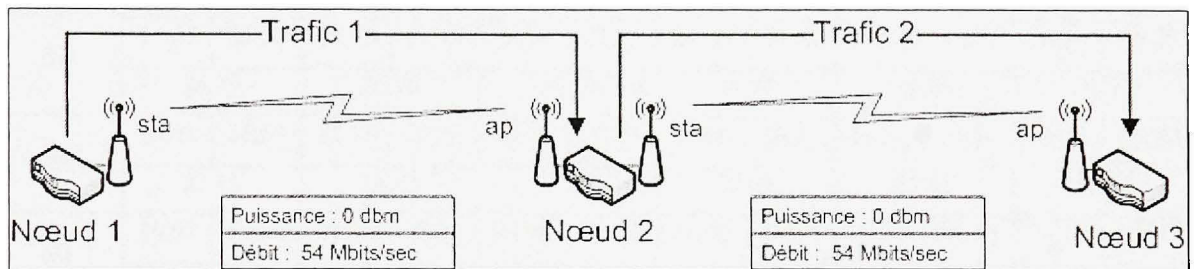


Figure 3.2 Topologie pour l'étude des interférences.

Tableau 3.2 Interférences 802.11a VS 802.11g (Mbps)

802.11a \ 802.11g	1		2		6		8		11	
48	27,50	12,79	27,17	19,45	27,23	7,94	28,44	2,03	27,65	8,10
	20,15		23,31		17,59		15,24		17,88	

Les résultats de la deuxième étude (Tableau 4.3) montrent l'existence d'interférence entre canaux proches du 802.11a. Ces interférences sont éliminées lorsque les canaux sont espacés de plus d'un canal. Quatre canaux peuvent ainsi être utilisés simultanément sans créer d'interférence. Pour la suite de l'étude, les canaux 40, 48, 56 et 64 sont utilisés.

Tableau 3.3 Interférences inter 802.11a (Mbps)

802.11a	36		40		44		48		52		56	
36	12,99	12,54	12,71	20,48	23,83	28,53	24,02	28,76	25,19	28,98	24,06	29,00
	12,77		16,60		26,18		26,39		27,09		26,53	
40	11,23	20,63	10,95	14,76	2,55	26,99	24,50	28,37	24,50	28,71	24,59	28,86
	15,93		12,86		14,77		26,61		26,73		26,73	
44	25,13	25,97	6,43	26,50	10,80	16,01	10,81	25,28	24,94	28,60	25,04	29,03
	25,55		16,47		13,41		18,05		26,78		27,04	
48	25,13	25,97	24,92	27,98	20,44	18,99	13,47	13,95	21,99	25,10	24,90	29,03
	25,93		26,45		19,72		13,71		23,55		26,97	
52	25,05	26,52	25,45	28,10	26,14	27,03	24,12	24,42	11,40	14,76	20,53	26,23
	25,79		26,78		26,59		24,27		13,08		23,38	
56	27,61	26,59	25,76	27,54	25,10	27,27	26,98	28,23	25,52	25,80	10,36	16,33
	27,10		26,65		26,19		27,61		25,66		13,35	
60	26,97	26,36	27,14	28,25	27,46	26,79	26,60	28,36	26,15	28,38	24,11	26,47
	26,67		27,70		27,13		27,48		27,27		25,29	
64	25,48	26,83	25,78	27,84	25,46	26,99	25,66	28,09	25,84	28,22	25,29	29,08
	26,16		26,81		26,23		26,88		27,03		27,19	

3.1.5 Synthèse

Cette étude de calibration a permis de fixer les paramètres de base du banc de test :

- Le débit à 54 Mbps pour une performance maximale.
- La puissance à 0 dbm pour une consommation d'énergie réduite.
- La norme 802.11a (5 GHz) pour le réseau dorsal, pour ses faibles interférences avec l'environnement extérieur et intérieur.
- La norme 802.11g (2,4 GHz) pour le réseau d'accès, pour sa grande popularité dans les systèmes mobiles.
- Pour le réseau dorsal, quatre canaux peuvent être utilisés simultanément sans créer d'interférences (40, 48, 56 et 64).

Tableau 3.4 Performance des liens "à vide" (Mbps)

	Traffics	Débit ascendant	Débit descendant
Ethernet	Séparés	51,16 +/- 0,3	51,25 +/- 0,23
	Simultanés	26,96 +/- 1,68	29,81 +/- 1,7
WiFi	Séparés	27,08 +/- 0,77	28,07 +/- 0,55
	Simultanés	15,67 +/- 3,01	11,51 +/- 3,07

La performance des liens point-à-point est synthétisée dans le Tableau 4.4. Elle est en fait inférieure aux attentes. Le lien Ethernet, configuré à 100 Mbps en full duplex offre une bande passante de 51 Mbps qui est partagée entre les trafics ascendants et descendants. Le lien WiFi, configuré à 54 Mbps offre une bande passante de 27 Mbps inégalement répartie entre les trafics montants et descendants. Ces écarts de performances sont dus à des limites intrinsèques du matériel. Pour la suite, ces valeurs correspondent aux performances maximales attendues des interfaces Ethernet et WiFi.

3.2 Performance des réseaux WiFi maillés

Une analyse générale de la performance des réseaux maillés est réalisée sur la topologie (Figure 4.1). Les limites et leurs origines sont présentées. Les paramètres étudiés sont l'impact du nombre de sauts, l'équité entre les MP, les effets de l'Internet, l'impact du taux de perte et le partage de la bande passante.

3.2.1 Effet des sauts

L'impact du nombre de sauts est étudié avec les quatre canaux 802.11a sans interférence. Des mesures des trafics montants et descendants sont réalisées successivement comme schématisées sur la Figure 4.3.

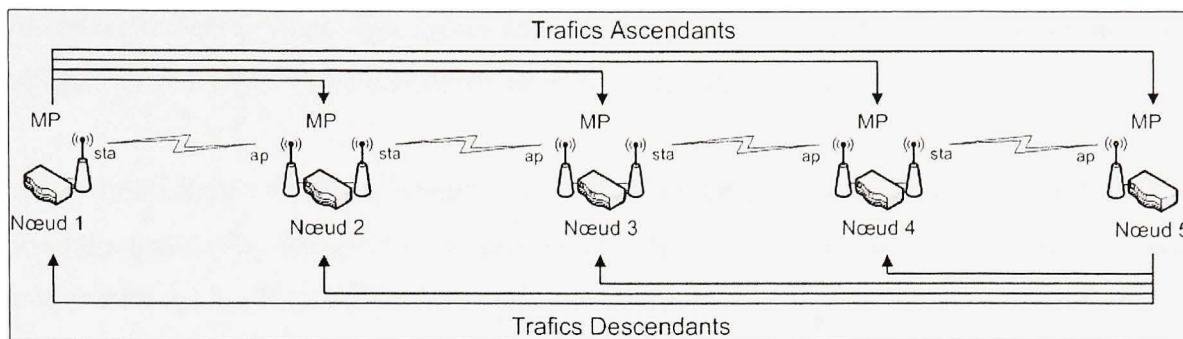


Figure 3.3 Topologie pour l'étude des sauts WiFi.

Les résultats présentés dans le Tableau 4.5 montrent une baisse de la bande passante avec l'augmentation du nombre de sauts. Cette baisse est due à une augmentation du taux de perte sur le trafic, associée à l'augmentation du nombre de saut; chaque saut WiFi est source d'erreur, le taux d'erreur augmente avec les sauts.

Le délai est réparti linéairement, il est inférieur à 0,4 ms par saut avec un total de 1,5 ms pour les quatre sauts.

Tableau 3.5 Impact du nombre de sauts (Mbps)

Nb. Sauts	1	2	3	4
Trafic ascendant	26,28 +/- 0,67	25,68 +/- 0,86	24,55 +/- 0,93	23,38 +/- 0,90
Trafic descendant	28,53 +/- 0,53	26,35 +/- 0,83	24,01 +/- 0,99	22,92 +/- 0,90

3.2.2 Équité, vue d'ensemble

Une étude générale de l'équité est menée sur la topologie 1 (Figure 4.1). Des trafics montants et descendants sont établis entre les nœuds MP et le nœud Internet.

Les résultats présentés sous forme de graphique sont utilisés pour illustrer le comportement général des trafics. La Figure 4.4 concerne les trafics ascendants, la Figure 4.5 les trafics descendants et la Figure 4.6 les trafics combinés. Dans les trois cas, la somme des trafics est

en moyenne de 28 Mbps. Les trafics sont regroupés et présentent un écart faible, de 5%; L'équité est correcte, la bande passante est répartie de façon homogène.

Les comportements étant équivalents dans les trois scénarios et la consommation de façon générale étant plus importante en provenance de l'Internet, la suite de l'étude portera uniquement sur le comportement des trafics descendants.

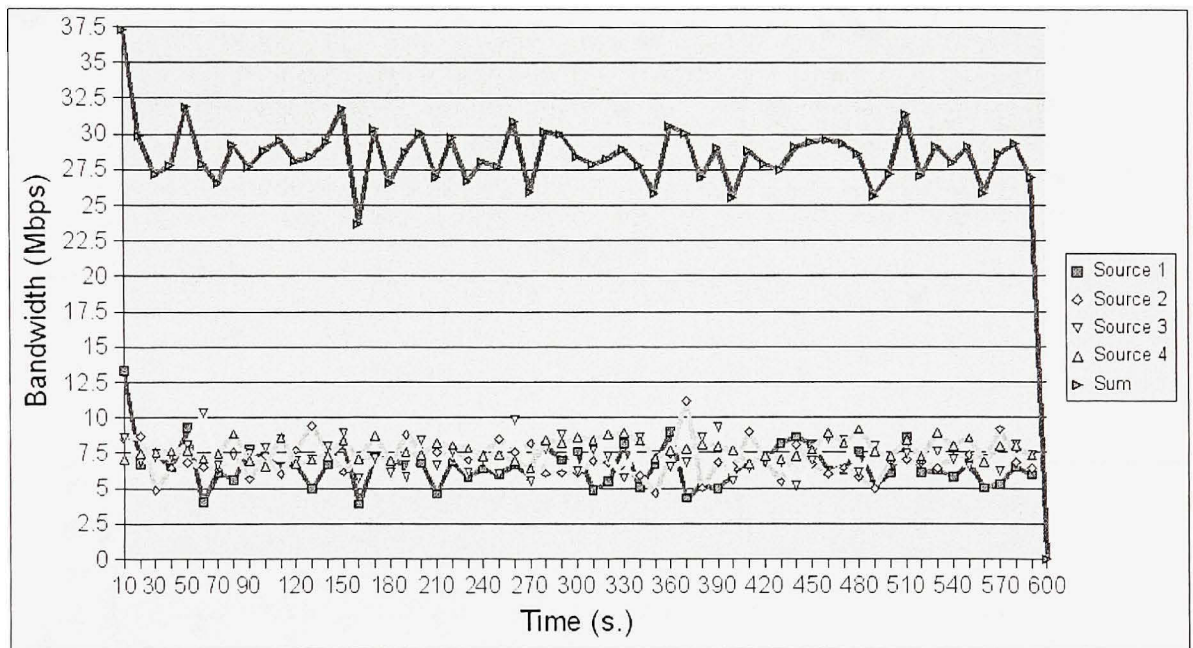


Figure 3.4 Trafics ascendants.

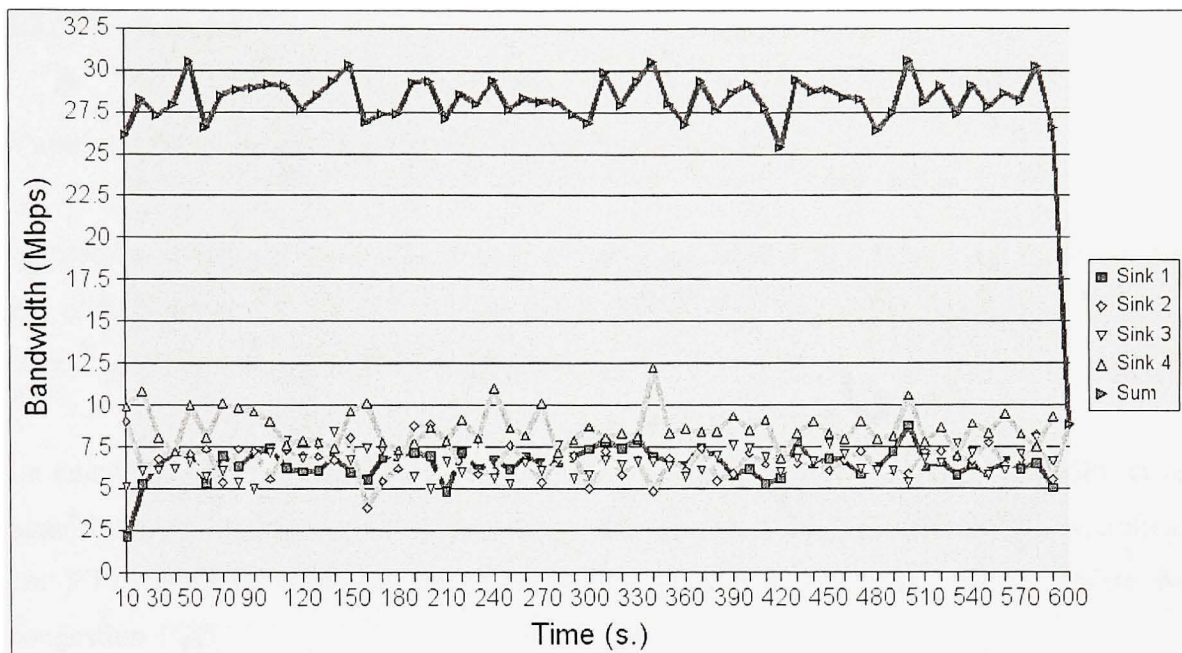


Figure 3.5 Trafics descendants.

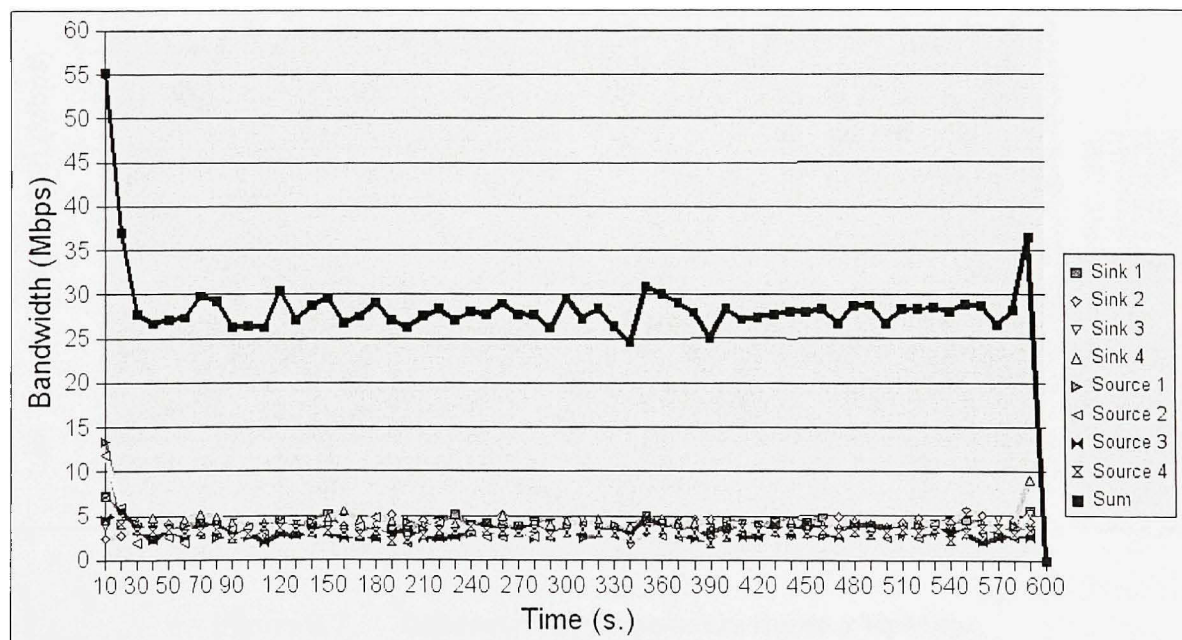


Figure 3.6 Trafics simultanés : ascendants et descendants.

3.2.3 Internet

L'impact de l'Internet sur la performance du réseau est étudié.

Le premier effet testé est la réduction de la bande passante à 10 Mbps ce qui correspond à une offre Internet « classique ». Le graphique (Figure 4.7) montre des résultats similaires à l'étude de l'équité sans goulot d'étranglement.

Le deuxième effet est l'impact du délai. 100 ms ont été ajoutées entre le nœud MPP et le nœud Internet. Le graphique (Figure 4.8) montre une nette baisse de la stabilité des trafics. Un RTT important implique des changements fréquents de la taille de la fenêtre de congestion TCP.

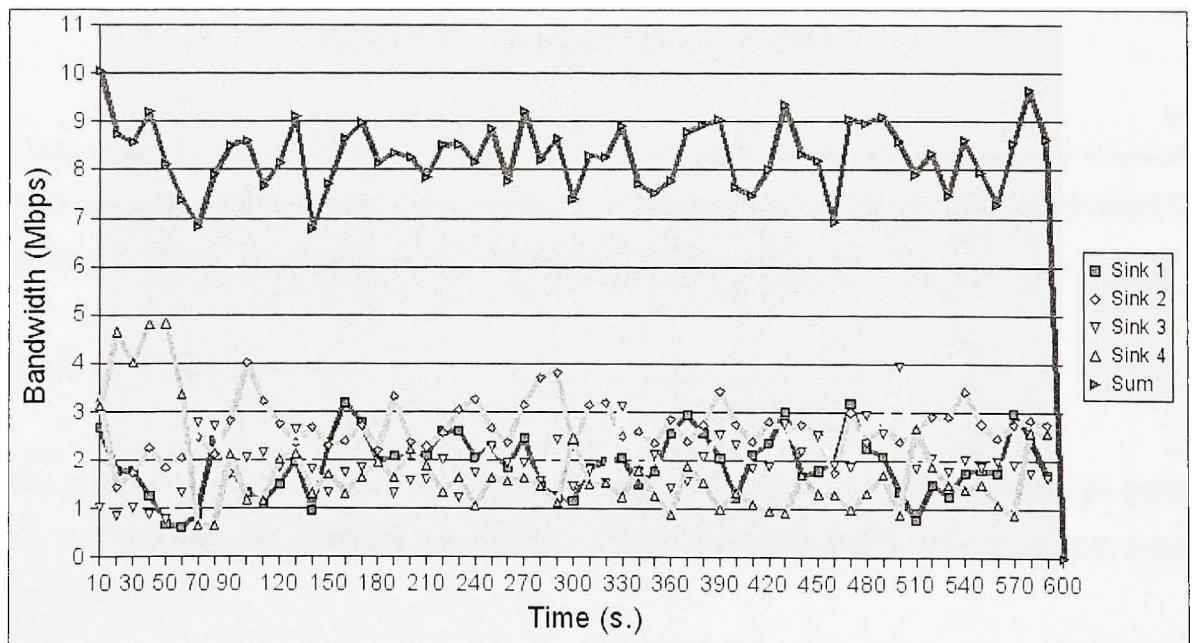


Figure 3.7 Internet : Bande passante limité à 10 Mbps.

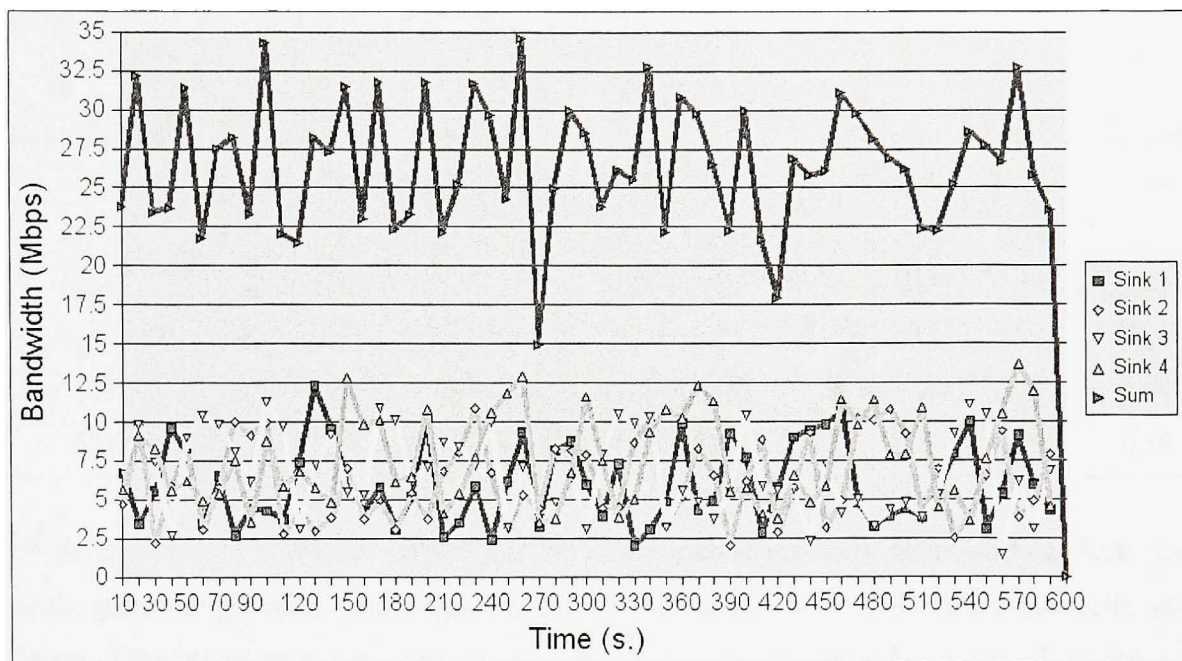


Figure 3.8 Internet : Délais de 100 ms.

Cette analyse a permis de montrer que l'Internet a un grand impact sur la stabilité des trafics mais un impact faible sur l'équité générale. Pour la problématique de cette étude, l'Internet à un impact mineur sur la performance des réseaux WiFi maillés.

3.2.4 Taux de perte

Les réseaux WiFi étant sujets aux aléas des conditions extérieurs, l'impact du taux de perte est étudié. Deux cas extrêmes sont testés : une perte uniformément répartie et une perte localisée de 5%.

Les résultats de la perte uniforme (Tableau 4.6) montrent un lien étroit entre l'équité et le taux de perte. À 0% de perte, l'équité est de 0,99, alors qu'à 5% elle est de 0,38. Plus les nœuds sont éloignés du MPP, plus ils sont sujets aux pertes et donc plus leur bande passante est faible.

Tableau 3.6 Équité VS Perte uniforme (Mbps)

Taux de perte	Nœud 1	Nœud 2	Nœud 3	Nœud 4	Somme	Équité
0%	6,44 +/- 0,87	6,55 +/- 0,89	6,38 +/- 0,87	8,21 +/- 0,96	27,59	0,99
0,1%	4,77 +/- 0,54	5,69 +/- 0,61	6,87 +/- 0,79	10,41 +/- 0,79	27,74	0,91
0,2%	4,68 +/- 0,47	5,51 +/- 0,61	7,08 +/- 0,63	10,49 +/- 0,82	27,76	0,91
0,5%	4,22 +/- 0,41	5,20 +/- 0,39	7,01 +/- 0,37	11,18 +/- 0,65	27,62	0,87
1%	3,25 +/- 0,38	4,56 +/- 0,52	6,85 +/- 0,52	12,29 +/- 0,76	26,94	0,79
5%	0,12 +/- 0,11	0,33 +/- 0,16	1,00 +/- 0,34	5,87 +/- 0,99	7,32	0,38

La perte localisée (Tableau 4.7) a pour effet une quasi-séparation du réseau en deux. La bande passante partagée en dessous du goulot d'étranglement est plus faible que celle au-dessus. Dans chacune de ces deux zones, l'équité est correcte entre les nœuds. L'équité est rétablie lorsque le goulot d'étranglement est en sortie (entre les nœuds 4 et 5).

Tableau 3.7 Équité VS Perte localisée (Mbps)

Localité de la perte	Nœud 1	Nœud 2	Nœud 3	Nœud 4	Somme	Équité
1-2	0,68 +/- 0,13	8,78 +/- 1,47	8,42 +/- 1,36	10,00 +/- 1,78	27,88	0,78
2-3	0,70 +/- 0,17	0,72 +/- 0,20	11,81 +/- 4,00	14,76 +/- 3,59	27,99	0,55
3-4	0,73 +/- 0,13	0,74 +/- 0,18	0,78 +/- 0,18	26,17 +/- 0,70	28,43	0,29
4-5	4,29 +/- 0,60	4,58 +/- 0,66	4,96 +/- 0,87	5,18 +/- 0,80	19,02	0,99

D'autre part, cette étude permet de mettre en valeur l'impact des pertes sur une connexion TCP. Lors de l'élaboration d'un réseau maillé, une attention particulière doit être portée au positionnement des nœuds et antennes pour réduire au maximum les sources d'erreur.

3.2.5 Partage de la bande passante

Les réseaux WiFi maillés accueillent un grand nombre de clients établissant chacun des trafics avec l'Internet. L'impact de ce partage de la bande passante est étudié par l'ajout de trafics entre l'Internet et les MP.

Un premier test de l'impact du partage de la bande passante est mené sur le réseau multisaut (Figure 4.1). Les résultats (Tableau 4.8) montrent l'apparition d'une perte de l'équité avec l'augmentation du nombre de trafics simultanés. Les MP les plus éloignés de l'Internet ont les débits les plus faibles.

Tableau 3.8 Équité VS Multiplication des trafics (Mbps)

Nb. Trafics	Nœud 1	Nœud 2	Nœud 3	Nœud 4	Somme	Équité
1	4,86 +/- 2,02	5,95 +/- 2,02	7,51 +/- 1,68	7,81 +/- 1,84	26,13	0,97
2	4,43 +/- 2,37	7,91 +/- 2,36	5,79 +/- 3,34	10,77 +/- 1,96	28,90	0,90
3	4,43 +/- 2,01	4,34 +/- 1,82	8,22 +/- 2,12	12,42 +/- 2,28	29,41	0,66

Les réseaux maillés alliant des structures en arbre, un deuxième test est mené sur la topologie en Figure 3.9. L'analyse porte sur le partage de la bande passante d'un point d'accès avec quatre clients. Les résultats (Tableau 4.9) démontrent une non-équité engendrée par un point d'accès WiFi. L'augmentation des trafics a pour conséquence une réduction de l'équité entre les trafics TCP.

Cette étude a démontré l'existence de limites de performance dans le partage de la bande passante avec un grand nombre de flux. La première est liée à la norme WiFi : elle survient lors de la saturation du point d'accès. La deuxième est liée au nombre de sauts avec l'augmentation du nombre de trafics. Les embranchements WiFi et les sauts sont ainsi des sources de non-équité.

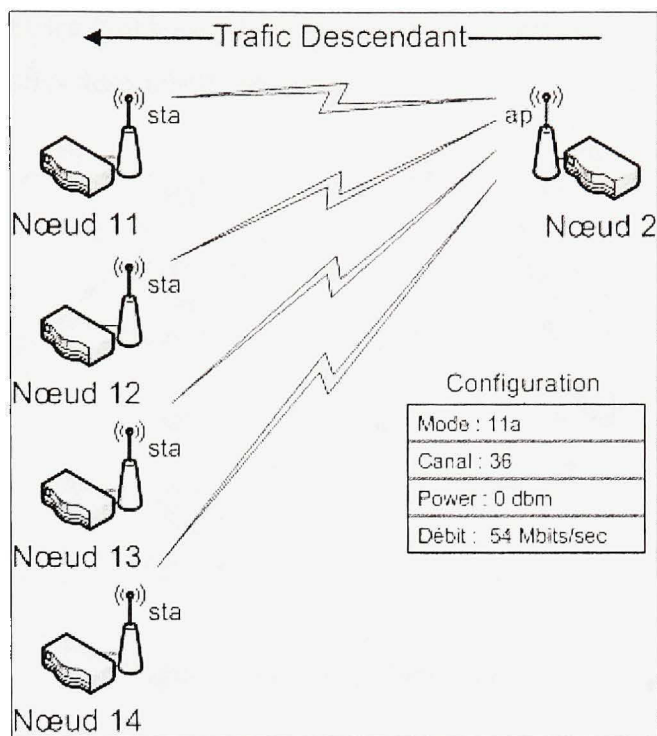


Figure 3.9 Topologie pour l'étude du partage de la bande passante des embranchements WiFi.

Tableau 3.9 Partage de la bande passante dans un point d'accès WiFi (Mbps)

Nb. Trafics par client	Nœud 11	Nœud 12	Nœud 13	Nœud 14	Équité
1	6,28 +/- 2,03	6,34 +/- 2,46	6,88 +/- 1,82	7,44 +/- 2,15	0,99
2	5,34 +/- 2,77	4,21 +/- 2,34	8,85 +/- 3,01	8,62 +/- 2,71	0,75
3	8,42 +/- 2,99	4,88 +/- 2,93	8,16 +/- 3,73	6,28 +/- 3,37	0,69

3.3 Mécanismes de QoS et performance

L'impact de mécanismes de QoS sur la performance est maintenant étudié. Une nouvelle topologie est mise en place (Figure 4.10) pour démontrer l'apport des mécanismes de QoS de niveau 3 proposés. Elle est composée des 3 acteurs : MP, MPP et Internet qui sont disposés

de manière à combiner les problèmes de sauts et d'embranchements WiFi. Les mesures sont effectuées sur trois trafics descendants par nœud.

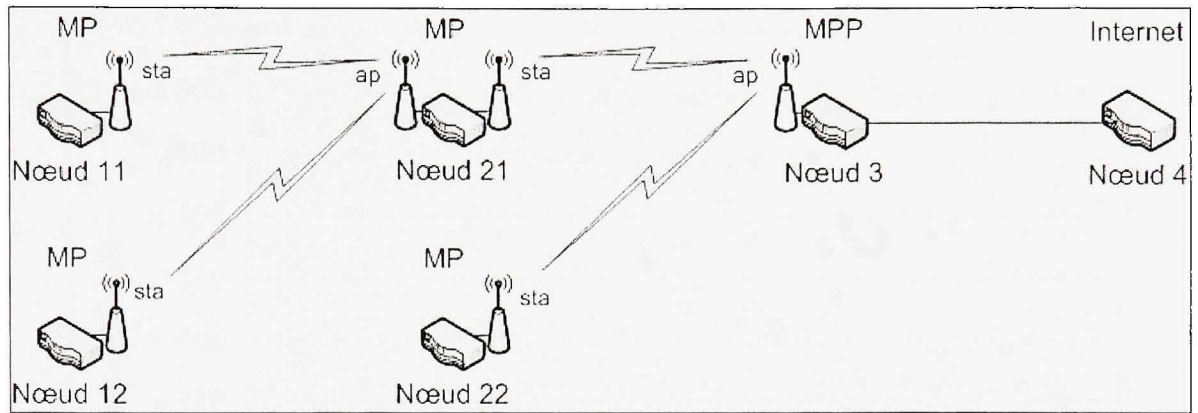


Figure 3.10 Topologie arbre.

Trois mécanismes de QoS sont présentés. Le premier est une limitation de la bande passante pour confiner l'effet de la congestion des points d'accès. Le deuxième est la mise en place de files d'attente à priorités égales (WFQ) entre les nœuds. Puis, le troisième est la mise en place de tunnels de taille fixe. Pour finir, une validation de ces mécanismes est réalisée par l'ajout d'une perte uniforme.

3.3.1 Limite en bande passante des points d'accès

Les analyses précédentes ont montré l'apparition d'une perte d'équité dans le partage de la bande passante d'un point d'accès. La topologie de réseau WiFi (Figure 4.9) est à nouveau utilisée. Le test porte sur la mise en place d'une limite en bande passante.

Le graphique en Figure 3.11 représente l'évolution de l'équité en fonction du niveau de la limite en bande passante. L'équité est conservée jusqu'à 25 Mbps.

Pour la suite de l'étude, une limite de bande passante à 25 Mbps est généralisée à l'ensemble des points d'accès.

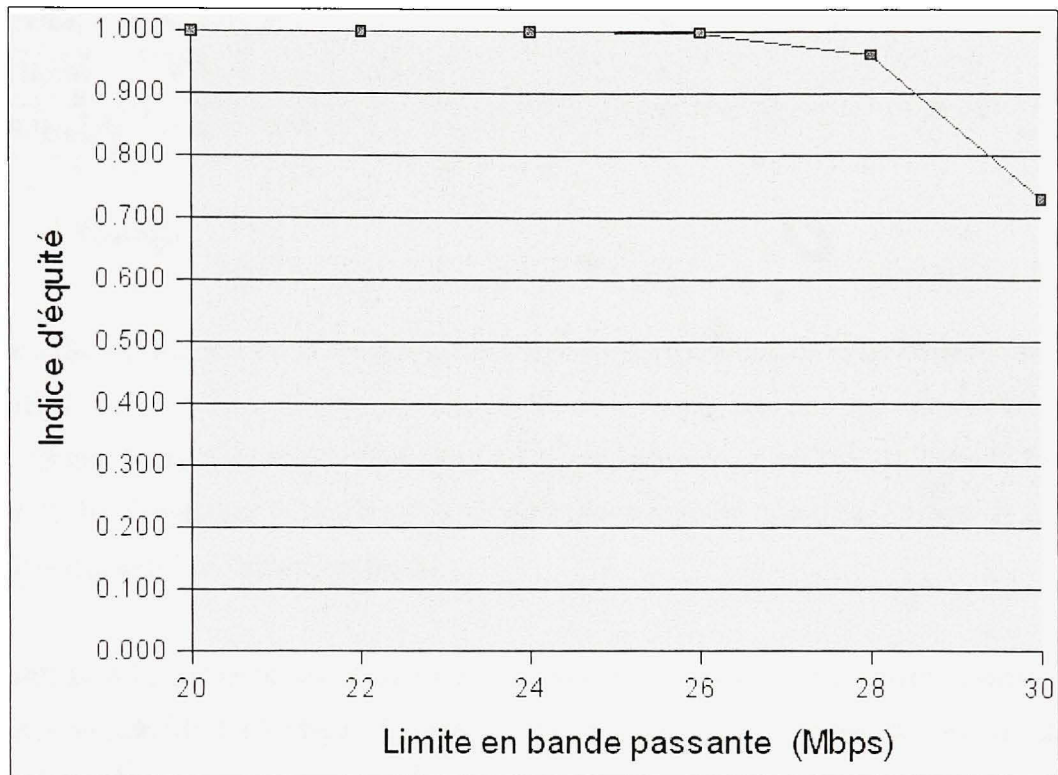


Figure 3.11 Équité VS limite en bande passante du point d'accès.

3.3.2 Files d'attente

Les files d'attente à priorité égale sont maintenant étudiées. Elles sont utilisées pour améliorer l'équité entre les nœuds. Des files d'attente parallèles sont attribuées à chaque nœud. Elles sont placées à l'entrée Internet, sur l'interface Ethernet du nœud 3.

Les résultats sont présentés dans le Tableau 4.10. L'utilisation des files d'attente combinée à la limite en bande passante des points d'accès améliore l'équité de 15%.

Tableau 3.10 Impact de la limite en bande passante et des files d'attente (Mbps)

	Nœud 11	Nœud 12	Nœud 21	Nœud 22	Somme	Équité
À vide	9,76 +/- 0,82	2,35 +/- 0,22	4,14 +/- 0,91	0,48 +/- 0,92	25,73	0,80
Ap limite	5,53 +/- 0,56	2,83 +/- 0,35	8,39 +/- 0,68	7,26 +/- 0,72	24,00	0,89
Ap limite, File	5,43 +/- 0,55	3,29 +/- 0,43	7,91 +/- 0,70	7,34 +/- 0,71	23,97	0,92

3.3.3 Tunnels

L'utilisation des tunnels est envisagée pour améliorer la répartition de la bande passante entre les trafics. Ils permettent de réserver une partie de la bande passante qui est associée à un trafic. Quatre tunnels de tailles identiques sont positionnés en parallèle sur l'entrée Internet (nœud 3). Ils remplacent la limite en bande passante des points d'accès. L'impact de la taille des files d'attente sur l'équité est étudié.

Le Tableau 4.11 présente les résultats de l'évolution de l'équité avec l'augmentation de la taille des tunnels (de 4 à 8 Mbps). L'équité reste constante à 1, néanmoins, des écarts entre les débits moyens des nœuds apparaissent à partir de 7 Mbps. La valeur optimale de la limite par tunnel est de 6 Mbps ce qui correspond à un total de 24 Mbps.

Tableau 3.11 Équité VS Taille du tunnel (Mbps)

Limite par tunnel	Nœud 11	Nœud 12	Nœud 21	Nœud 22	Somme	Équité
4 Mbps	3,85 +/- 0,13	3,86 +/- 0,18	3,86 +/- 0,18	3,86 +/- 0,24	15,43	1
5 Mbps	4,83 +/- 0,32	4,83 +/- 0,24	4,83 +/- 0,26	4,83 +/- 0,23	19,32	1
6 Mbps	5,79 +/- 0,29	5,80 +/- 0,30	5,79 +/- 0,36	5,78 +/- 0,36	23,16	1
7 Mbps	6,42 +/- 0,42	6,44 +/- 0,36	6,42 +/- 0,36	6,67 +/- 0,33	25,95	1
8 Mbps	6,45 +/- 0,48	6,45 +/- 0,40	6,45 +/- 0,54	6,89 +/- 0,47	26,24	1

3.3.4 Pertes et mécanismes de QoS

Les mécanismes présentés précédemment sont maintenant testés dans des conditions dégradées. Un taux de perte uniforme de 1% est ajouté dans le réseau.

Le résultat des quatre scénarios est présenté dans le Tableau 4.12. Même dans des conditions dégradées les files d'attente et tunnels ont un impact positif sur l'équité. Le mécanisme le plus performant est là encore le tunnel avec un indice d'équité de 1.

Tableau 3.12 Mécanismes de QoS et perte uniforme de 1% (Mbps)

	Nœud 11	Nœud 12	Nœud 21	Nœud 22	Somme	Équité
À vide	5,04 +/- 0,33	5,00 +/- 0,36	7,88 +/- 0,38	8,59 +/- 0,50	26,52	0,94
Ap Limite	4,33 +/- 0,42	4,49 +/- 0,56	7,55 +/- 0,65	7,69 +/- 0,66	24,05	0,93
Ap Limite, Files	4,70 +/- 0,45	4,61 +/- 0,44	7,37 +/- 0,49	7,38 +/- 0,67	24,06	0,95
Ap Limite, Tunnels (6Mbps)	5,67 +/- 0,22	5,67 +/- 0,28	5,74 +/- 0,29	5,73 +/- 0,23	22,81	1,00

3.4 Discussions

Les analyses précédentes sur les réseaux maillés ont permis une série d'observations sur la performance. S'en est suivi une étude des mécanismes de QoS permettant de résoudre en partie ces problèmes. Ces observations sont maintenant synthétisées, ainsi que l'impact des mécanismes de QoS et nous poursuivons par une discussion sur une mise en place effective de ces mécanismes.

3.4.1 Limites et observations

RTT : Le délai dans les réseaux maillés est faible. L'impact du réseau sur le RTT est faible comparé à celui de l'Internet.

Pertes : Le taux de perte a un impact direct et important sur l'équité. TCP est très sensible aux pertes, sa performance peut être largement dégradée par des connexions WiFi dégradées.

Sauts : Le partage de la bande passante entre un grand nombre de trafics a pour effet une perte d'équité entre les sauts WiFi.

Embranchements : L'augmentation des clients WiFi sur un point d'accès a pour effet un partage inégal de la bande passante.

3.4.2 Mécanismes de QoS

Limite en bande passante : Pour éviter la saturation des points d'accès, une limite en bande passante est fixée. Elle permet de résoudre la perte d'équité au niveau des embranchements.

Files d'attente : Les files d'attente permettent des améliorations notables de l'équité. Mais elles impliquent des délais supplémentaires sur le réseau.

Tunnels : Les tunnels, mécanisme de QoS le plus efficace, permettent à la fois de stabiliser et d'équilibrer les trafics.

3.4.3 Exploitation en environnement réel

Les mécanismes de QoS testés sont simples à mettre en œuvre et les améliorations sont significatives. Néanmoins, cette mise en place expérimentale avait pour but de tester les limites. Elle ne correspond pas à une problématique d'ingénierie de réseau plausible. C'est ainsi qu'une politique de gestion de la QoS est proposée dans les recommandations de ce mémoire.

3.5 Conclusion

Dans ce chapitre un tour d'horizon de la performance des réseaux maillés multiradio a été présenté. Des limites dans la gestion de l'équité ont été identifiées. Des mécanismes de QoS de niveau 3 ont été testés et ont permis d'améliorer la performance et de rétablir l'équité. Cette analyse a permis de prouver qu'il est possible d'améliorer efficacement la performance des réseaux maillés en jouant uniquement sur des paramètres de niveau 3.

La mise en place d'une politique de gestion de la QoS rudimentaire permet d'accroître de façon considérable la performance dans les réseaux maillés sans changement technologique. Une stratégie envisageable de mise en place en milieu réel est proposée plus loin, dans les recommandations.

D'autres travaux sont envisagés dans le test de la performance du brouillon 802.11s. Ce protocole émergent de niveau 2 doit permettre une amélioration de l'équité ce qui sera vérifié. Des ouvertures pour la mise en place effective du protocole pourraient alors être développées.

CHAPITRE 4

MODÈLE D'ABSTRACTION

Les étudiants en réseaux et télécommunications d'aujourd'hui sont de moins en moins familiers avec l'environnement Unix. La philosophie Unix, très différente de celle de Windows, les rend incapables d'utiliser des équipements Unix. Ils doivent passer de longs mois de formation avant d'être capables de configurer des expériences réseau sur un banc de test.

Pour remédier à cela, nous proposons une approche abstraite de l'environnement expérimental pour la plate-forme. Elle permet d'ouvrir l'expérimentation sur banc de test à un plus large public. Ces nouveaux points de vue devraient permettre d'apporter de nouvelles perspectives de recherche.

C'est ainsi qu'est développé un modèle d'abstraction pour l'élaboration d'expériences dans l'interface graphique de la plate-forme. Le vocabulaire choisi est celui du jeu pour sa popularité et ses similitudes avec l'exécution d'expérience. Une expérience devient ainsi un ensemble de déplacements de joueurs en suivant des règles sur un plateau de jeu.

Ce chapitre commence par une présentation de l'approche du modèle du jeu. Puis, il présente les sept niveaux du modèle. Enfin, une discussion et une conclusion sont proposées.

4.1 Approche

Le modèle d'abstraction est une vision pratique des expériences. Il a été développé pour permettre à des utilisateurs novices de réaliser des expériences de manière intuitive. Il est dissocié de l'univers Unix et basé sur le vocabulaire du jeu. Le modèle comprend sept niveaux répartis en quatre points de vue : environnement, banc de test, exécution et résultat (Tableau 3.1).

Tableau 4.1 Modèle d'abstraction

Résultat	7. Score
Exécution	6. Partie
Banc de test	5. Déplacement 4. Règle 3. Plateau
Environnement	2. Joueur 1. Pion
Unix	

L'environnement est une représentation virtuelle des nœuds et services. Le banc de test est la définition de l'expérience : réseau, simulation et trafics. L'expérience est la définition du scénario et l'exécution. Le résultat est le traitement des données en résultats.

4.2 Niveaux

Les sept niveaux sont maintenant présentés. Pour mieux comprendre leur fonctionnement dans le modèle, la présentation est accompagnée d'un exemple de test de performance. La problématique étudiée est l'impact des pertes d'un réseau sur une communication client-serveur. Trois nœuds Alix2d3 sont installés en ligne. Le nœud 1 est utilisé comme client TCP, le nœud 2 comme routeur simulant un effet de perte et le nœud 3 comme serveur TCP.

4.2.1 Pion

Un pion représente un service. Il est caractérisé par des commandes d'initialisation et d'arrêt qui permettent l'exécution et l'arrêt du service. Le type d'exécution est spécifié par un « synchrone » ou « asynchrone ». Les paramètres sont la liste des éléments variables du service. Ils sont configurés dans les couches supérieures du jeu. Les services de mesures sont associés à un script d'extraction des données permettant d'extraire les données des fichiers

traces à la fin de l'expérience. Chaque service est associé au niveau du jeu dans lequel il est joué. Cinq sont définis :

- Plateau Connexion : pions de configuration d'interfaces.
- Plateau Route : pions de routage.
- Règle Simulation : pions de simulation réseau.
- Déplacement Initialisation : pions de mesure ou d'initialisation du déplacement.
- Déplacement Mouvement : pions de mise en place de trafic.

Tableau 4.2 Pions 1/2

Pions	Interf Ethernet	Route
Cmd init.	"ifconfig create \${interfPhy} \${interfIp}"	"route add -net \${netIp} -netmask \${netMask} -host \${hostIp}"
Cmd arrêt	"ifconfig \${interf} destroy"	route del -net \${netIp}
Type exéc.	sync	sync
Paramètres	interfPhy=[eth1,eth2] interfIp=""	netIp="" netMask="" hostIp=""
Script extr. données	-	-
Type de pion	Plateau-Connexion	Plateau-Route

Tableau 4.3 Pions 2/2

Pions	Tx Perte	TCP Puits	TCP Source
Cmd init.	"ipfw \${interf} plr \${lostRate} src-ip any dst-ip any"	"iperf -s"	"iperf -c \${destIp} -t \${time} - i \${interv}"
Cmd arrêt	"ipfw \${interfPhy} destroy"	Kill	Wait
Type exéc.	sync	async	async
Paramètres	interfPhy="" lostRate=""	-	destIp="" time="60" interv="1"
Script extr. données	-	-	iperfcExtractor.sh -c \${destIp} -t \${time} -i \${interv}
Type de pion	Règle	Déplacement-Initialisation	Déplacement-Mouvement

Pour l'analyse de la performance de TCP, cinq pions sont créés (Tableau 3.2 et Tableau 3.3). Un premier pion « Interf Ethernet » permet la création et suppression des interfaces Ethernet. Le pion « Route » permet l'ajout dans la table de routage de chemins vers des réseaux distants. Le pion « Tx Perte » est le service simulation de taux de perte. Les pions « TCP Source » et « TCP Puits » permettent la mise en place d'un trafic TCP de la source vers le puits.

4.2.2 Joueur

Un joueur est la représentation virtuelle d'un nœud (Tableau 3.4). Il est caractérisé par les pions (services) supportés. Pour une meilleure lisibilité, les pions sont organisés par niveaux de jeu.

Tableau 4.4 Joueurs

Joueurs	Client/Serveur Eth	Router Eth avec Pertes
Déroulement	- TCP Puits - TCP Source	-
Règles	-	- Tx Perte
Plateau	- Route - Interf Ethernet = "Interf1"	- Interf Ethernet = "Interf1" - Interf Ethernet = "Interf2"

Pour l'analyse de la performance, deux joueurs sont définis. Le « client/serveur » est utilisé pour les nœuds 1 et 3; il permet la configuration d'une interface Ethernet, la définition des routes dans les tables de routages et la mise en place des sources et puits du trafic TCP. Le « Routeur » est défini pour le nœud 2; il permet la configuration de deux interfaces Ethernet et l'ajout d'un taux de perte variable.

4.2.3 Plateau

Le plateau est la définition du réseau et des routes (Tableau 3.5). Les joueurs sont positionnés sur le plateau, un nom unique leur est attribué. Les pions Plateau Connexion et Plateau - Route sont configurés pour chaque joueur. Les paramètres expérimentaux du plateau sont les paramètres utilisés dans le scénario de l'expérience au niveau partie.

Tableau 4.5 Plateau

Plateau	3 nœuds en ligne		
Noms J.	Noeud1 = Client/Serveur Eth	Noeud2 = Router Eth avec Perte	Noeud3 = Client/Serveur Eth
Plateau-Cnx	Interf1 - interfIp = "192.168.1.2" Interf1 - interfPhy = "eth1"	Interf1 - interfIp = "192.168.1.1" Interf1 - interfPhy = "eth1" Interf2 - interfIp = "192.168.2.2" Interf2 - interfPhy = "eth2"	Interf1 - interfIp = "192.168.2.1" Interf1 - interfPhy = "eth1"
Plateau-Rtes	Route - netIp = "192.168.2.0" Route - netMask = "255.255.255.0" Route - hostIp = "192.168.1.1"	-	Route - netIp = "192.168.1.0" Route - netMask = "255.255.255.0" Route - hostIp = "192.168.2.2"
Para.Exp	-		

Les joueurs « client/serveur » et « routeur » sont positionnés sur le plateau « 3 nœuds en ligne ». Les joueurs sont nommés noeud1, noeud2 et noeud3. Deux réseaux sont configurés, le 192.168.1.0 entre les noeud1 et noeud2 et le 192.168.2.0 entre les noeud2 et noeud3. Les routes pour accéder aux réseaux distants sont ajoutées dans les noeud1 et noeud3. Aucun paramètre expérimental n'est défini, car la problématique de l'analyse n'implique pas d'éléments de configuration réseau.

4.2.4 Règle

Une règle est la définition des éléments de simulation ajoutés sur le banc de test (Tableau 3.6). Elle est associée à un et un seul banc de test. Les pions Règle Simulation sont définis pour chaque joueur. Les paramètres expérimentaux sont utilisés lors de la définition du scénario de la partie.

La règle « Taux de perte routeur » est définie pour l'expérience. Le pion « Tx de perte » est mis en place sur l'interface eth1 du noeud2. Le taux de perte est défini comme paramètre expérimental; sa valeur est configurée dans le scénario de la partie.

Tableau 4.6 Règle

Règle	Taux de perte routeur		
Plateau	3 nœuds en ligne		
Joueurs	Noeud1	Noeud2	Noeud3
Règles-Simulation	-	Tx Perte – interfPhy = "eth1" Tx Perte – lostRate = $\${lostRate}$	-
Paramètres Expérimentaux	lostRate= " "		

4.2.5 Déplacement

Un déplacement correspond à l'activité temporelle de l'expérience (Tableau 3.7). Il comporte la mise en place des pions Déplacement Initialisation et des pions Déplacement Mouvement sur le plateau. Pour le plateau « 3 nœuds en ligne » un déplacement de trafic ascendant (noeud1 vers noeud3) est défini. Il est nommé « Mesure de l'évolution du trafic montant ». Les durées du trafic et des intervalles de mesures sont définies comme paramètres expérimentaux. L'expérience peut ainsi être mise en place une première fois avec une durée plus courte pour une évaluation de comportement. Le script de résultat est utilisé à la fin de l'expérience pour mettre en forme les données en résultat visuelles. Dans l'exemple, le script est utilisé pour créer un graphique de l'évolution temporelle du trafic.

Tableau 4.7 Déplacement

Déroulement	Mesure de l'évolution du trafic ascendant		
Plateau	3 nœuds		
Joueurs	Noeud1	Noeud2	Noeud3
Déplacement- Initialisation	-	-	TCP Puits
Déplacement- Mouvement	TCP Source - destIp = "192.168.2.2 " TCP Source - time = \${time} TCP Source - interv = \${interv}	-	-
Paramètres Expérimentaux	time = "" interv = ""		
Scripts de résultat	graphicTrafficEvolution.sh Noeud1 = "EvolutionTCPAscendant"		

4.2.6 Partie

Sur un même plateau, il est possible de définir une multitude de combinaisons de règles et de déplacements. La partie permet de définir l'itération de l'expérience exécutée. La partie « Analyse de l'impact du taux de perte sur une communication Ethernet » est créée. Elle associe la règle et le déplacement précédemment définis (Tableau 3.8). Les associations joueurs nœuds sont utilisées pour lier les joueurs virtuels aux nœuds réels. Le scénario définit les cinq valeurs de taux de perte (0%, 0.5%, 1%, 5% et 10%) analysé, la durée du trafic TCP à 10 minutes (600 secondes) et l'intervalle de mesure à 10 secondes.

Une fois la définition terminée, le script d'expérience (présenté dans les annexes) est automatiquement créé. L'expérience est exhaustive, toutes les possibilités sont évaluées grâce à l'utilisation des boucles. Le test est mis en place cinq fois pour les cinq taux de perte.

L'exécution terminée, les données et résultats du déplacement sont automatiquement extraits des traces. Pour cela les scripts pion, puis les scripts déplacement sont exécutés.

Tableau 4.8 Partie

Partie	Analyse de l'impact du taux de perte sur une communication Ethernet
Plateau	3 nœuds en ligne
Règle	Taux de perte routeur
Déplacements	Mesure de l'évolution du trafic ascendant
Joueurs – Nœuds	Noeud1="00 :0d :b9 :15 :bc :70" Noeud2="00 :0d :b9 :15 :ba :e4" Noeud3="00 :0d :b9 :15 :c3 :78"
Scénario	lostRate = [0,0.005,0.01,0.05,0.10] time = 600 interv = 10

4.2.7 Score

Un score est une définition de résultats de haut niveau (Tableau 3.9). Il synthétise le comportement d'un déplacement dans plusieurs parties. Le score « Trafic montant vs Taux de perte » permet par exemple de générer un graphique de l'évolution du trafic moyen en fonction du taux de perte. Le trafic moyen est calculé sur l'ensemble des parties jouées. Par exemple, la partie « Analyse de l'impact du taux de perte sur une communication Ethernet » a été exécutée trois fois. Les débits moyens des trois itérations sont synthétisés dans un graphique à barres. La mise en forme du score est réalisée par le script score.

Tableau 4.9 Score

Score	Trafic ascendant VS Taux de perte
Plateau	3 nœuds en ligne
Déplacement	Mesure de l'évolution du trafic ascendant
Script score	GraphiqueTraficMoyenVSTxPerte.sh

4.3 Discussion

Le modèle d'abstraction présenté est un apport substantiel pour la recherche. Il permet de simplifier la mise en place d'expériences réseau par un vocabulaire « universel », celui du jeu. Il ouvre ainsi l'élaboration d'expériences réseau à une nouvelle génération de chercheurs.

Le regroupement des actions par niveau permet de réduire grandement le travail du chercheur. Il ne travaille plus uniquement à bas niveau à l'élaboration de listes de commandes. Il travaille avec des joueurs et pions qu'il configure par niveau de jeu.

Les données qui sont réparties sur les nœuds sont automatiquement centralisées et traitées. Le chercheur définit des scores et suit leur évolution au travers des parties jouées.

4.4 Conclusion

Ce chapitre a présenté le modèle d'abstraction proposé pour rendre le développement d'expériences réseau sur la plate-forme plus intuitif. Le vocabulaire du jeu a été choisi pour sa large popularité et les fortes similitudes entre le déroulement d'une partie et d'une expérience.

L'exemple de la mise en place d'un test de performance d'une connexion TCP entre un client et un serveur au travers d'un routeur générant des pertes a permis d'illustrer le

fonctionnement du modèle. Il paraît plus facile de configurer des boîtes dont les entrées sont spécifiées qu'un script sans structure.

L'élaboration d'expérience est ainsi partagée en quatre points de vue : environnement, banc de test, exécution et résultat. La configuration de l'environnement préalable à l'élaboration d'expérience est la seule étape nécessitant des connaissances Unix. Les points de vue banc de test, exécution et résultat sont des empilements et configurations de boîtes. Ceci devrait permettre d'ouvrir l'élaboration à un plus large public; des étudiants sans connaissance Unix seront maintenant en mesure de réaliser des expériences réseau sur des équipements Unix.

CHAPITRE 5

DISCUSSION

Ce mémoire a permis l'étude sur banc de test de la performance des réseaux maillés multiradio. Une plate-forme a été développée pour automatiser l'exécution d'expériences sur banc de test. Un modèle d'abstraction basé sur le vocabulaire du jeu a été proposé pour rendre la plate-forme intuitive. L'étude de la performance a identifié des limites dues au protocole WiFi (IEEE 802.11) qui ont été corrigées par des mécanismes de QoS de niveau 3.

La plate-forme a été développée dans une approche modulaire, basée sur l'existant. Les expériences sont définies par un script d'exécution et des scripts de traitement des données en résultats. Les nœuds de test sont des équipements réseau qui se connectent au serveur central et attendent des commandes à exécuter. La plate-forme synchronise l'exécution des commandes sur les nœuds de test dans le cadre d'une expérience. Elle offre à la recherche un nouvel outil simple permettant de déployer rapidement des expériences sur banc de test. Les nouvelles technologies et protocoles pourront être validés plus rapidement. Dans le futur, il pourrait être intéressant de rapprocher les études sur banc de test des études en simulation; les scripts NS-2 pourraient être transcrits automatiquement en scripts exécutables par la plate-forme ou vice et versa.

Les réseaux maillés multiradio ont montré des limites de performance dans la gestion de l'équité au travers des embranchements et sauts WiFi. Lorsqu'un point d'accès WiFi subit une forte charge de trafics, le partage du média n'est plus équitable; aléatoirement, des clients ont plus de bande passante que d'autres. Les sauts WiFi causent des écarts de performance qui sont dus à une répartition de la bande passante gérée localement. Chaque routeur maillé partage sa bande passante avec les routeurs directement connectés à lui même; plus un client sera éloigné de la passerelle, plus sa part de bande passante sera partagée et donc réduite par rapport aux autres. Ces limites de performance sont dues à des limites intrinsèques du protocole WiFi.

Le modèle d'abstraction a été présenté dans le but d'ouvrir l'utilisation de la plate-forme à un plus large public. Il développe une approche ludique dans l'élaboration d'expériences réseau basée sur le vocabulaire du jeu. Une expérience est définie par un plateau de jeu sur lequel des joueurs déplacent de pions en suivant des règles. Les scripts d'exécution et de traitement des données sont automatiquement générés à partir de ces informations. Cette approche permet à des utilisateurs sans connaissance Unix d'élaborer des expériences sur banc de test. Ainsi, les nouveaux chercheurs moins familiers avec Unix n'auront plus besoin de suivre des formations longues et complexes pour être capable de réaliser leurs expériences. Ce qui devrait permettre d'apporter un regard neuf, non influencé par la philosophie Unix. Néanmoins, son fonctionnement et son apport à la plate-forme et au milieu de la recherche doivent encore être validés par sa mise en situation.

Pour rétablir l'équité, des mécanismes de QoS de niveau 3 ont été utilisés. Pour nos études, une limite en bande passante à 25 Mbps au niveau des points d'accès a permis de limiter les problèmes liés à leur saturation. Des files d'attente à priorités égales au niveau de la passerelle ont permis d'améliorer l'équité de 15%. Des tunnels au niveau de la passerelle ont permis de contrôler la répartition de la bande passante entre trafics et de les stabiliser. Ces mécanismes ont démontré qu'il est possible de réduire l'impact des limites intrinsèques du protocole WiFi en jouant uniquement sur des paramètres de niveau 3.

Les mécanismes de QoS présentés dans ce mémoire ne sont pas utilisables tels quels dans un déploiement réel. Nous proposons ainsi un déploiement plausible de ces mécanismes de QoS. La limite en bande passante des points d'accès est généralisée à tous les routeurs pour éviter les effets de saturation. La gestion de l'équité et de la répartition de bande passante est réalisée par une politique de QoS simple. Trois niveaux sont définis :

- Long terme, non prioritaire : pour les flux importants, mais non prioritaires (youtube, bittorrent...).
- Prioritaire : pour les flux prioritaires (VoIP, vidéo...).
- Autres : pour tous les autres flux (messagerie, nouvelles, réseaux sociaux...).

À chacun de ces niveaux est allouée une bande passante réservée par un tunnel. La classification des flux peut se faire simplement en se basant sur les ports ou la durée de vie des trafics. Des files d'attente à priorités égales sont insérées dans chacun des tunnels pour maintenir l'équité entre les clients par distinction d'adresses IP.

Il serait maintenant possible de pousser plus loin ces analyses. Le nombre de nœuds pourrait être multiplié pour tester la performance de nouvelles topologies. Cela pourrait mettre en avant de nouvelles limites dans la gestion de l'équité.

Les mécanismes de QoS pourraient être testés dans ces nouvelles topologies. Il est alors possible que l'approche centralisée sur la passerelle ne soit plus suffisamment efficace pour rétablir l'équité. Une approche décentralisée pourrait être envisagée et testées.

La performance des réseaux maillés et mécanismes de QoS de niveau 2 devrait être étudiée et comparée à celle de ceux de niveau 3. Cela permettrait de valider l'approche des réseaux maillés multiradio de niveau 3. D'autre part il serait possible de déterminer les limites des réseaux maillés de niveau 2 pour leurs améliorations futures.

CONCLUSION

Ce mémoire a permis dans le cadre de l'étude de la performance des réseaux maillés multiradio le développement d'une plate-forme d'exécution d'expériences réseau sur banc de test. Pour ouvrir son utilisation à des utilisateurs sans connaissance Unix, un modèle d'abstraction se basant sur le vocabulaire du jeu a été élaboré.

L'étude des perspectives de performance des réseaux maillés multiradio sur banc de test a été menée sur les topologies en ligne et en arbre. Les résultats ont montré des limites dans la répartition de la bande passante au niveau des sauts et embranchements WiFi. Des mécanismes de QoS simples de niveau 3 ont été testés : limite en bande passante des points d'accès, files d'attente et tunnels. Ils ont permis des améliorations notables de l'équité. Des études complémentaires devraient maintenant être menées pour valider ces mécanismes à plus large échelle, avec plus de nœuds maillés et dans des topologies plus complexes. D'autre part, des tests sur la performance des mécanismes de QoS de niveau 2 tel que 802.11e pourraient être envisagés pour être comparés avec ceux de niveau 3.

La plate-forme développée a permis d'automatiser les étapes de mise en place d'expériences réseau sur banc de test ce qui a permis un gain considérable de temps au niveau de la partie expérimentation de ce mémoire. L'approche modulaire et ouverte basée sur l'utilisation de scripts a été développée de façon à prévoir les évolutions des technologies futures ; tout type d'équipement, d'outil et de service peuvent être ajouté à la plate-forme. L'exécution des expériences est entièrement contrôlée par la plate-forme, les traces d'exécution automatiquement centralisées et traitées en données lisibles. La plate-forme en est aujourd'hui (juin 2010) à sa version 1.0. La partie interface graphique n'a pas encore été développée. Elle devrait faire partie des futures évolutions. D'autre part, il est prévu de proposer prochainement la plate-forme à la communauté open source.

Le modèle d'abstraction a été proposé comme une ouverture vers une nouvelle approche dans l'élaboration d'expériences sur banc de test permettant une utilisation par un plus large

public. Le vocabulaire du jeu a été choisi pour sa large popularité; l'élaboration d'un script d'expérience est intuitive. Les éléments du jeu sont positionnés et configurés graphiquement dans une interface Web. Cette nouvelle approche devrait faire partie des développements futurs de l'interface graphique Web. Son apport dans l'élaboration d'expériences réseau pourra alors être validé auprès des chercheurs.

Ce mémoire offre ainsi une plate-forme innovante dans la mise en place d'expérience réseau. L'utilisation d'équipements et technologies existantes est simplifiée par une élaboration de jeu graphique. La plate-forme a permis de démontrer la performance effective des réseaux maillés multiradio et de proposer une politique d'ingénierie de trafic basée sur des mécanismes simples de QoS.

ANNEXE I

Script d'expérience : performance d'une connexion Ethernet

```
1 #--Variables
2 Noeud1=00:0d:b9:15:bc:70
3 Noeud2=00:0d:b9:15:ba:e4
4
5 #-----Réseau-----
6 ${Noeud1} , sync , ifconfig eth1 192.168.1.1 up
7 ${Noeud2} , sync , ifconfig eth1 192.168.1.2 up
8
9 #-----Outils de simulation-----
10 #--Boucle
11 FOR lostRate IN [0,0.05,0.10]
12 {
13   ${Noeud2} , sync , ipfw eth1 plr ${lostRate} src-ip any dst-ip any
14
15   #-----Trafics-----
16   ${Noeud2} , async , iperf -s , id1
17
18   #--Blocs
19   Ascendant ::
20   {
21     ${Noeud1} , async , iperf -c 192.168.1.2 -t 600 -i 10 , id2
22     ${Noeud1} , wait , id2
23   }
24
25   ${Noeud2} , kill , id1
26
27   #-----Outils de simulation (relachement)-----
28   ${Noeud2} , sync , ipfw eth1 destroy
29 }
30
31 #-----Réseau (relachement)-----
32 ${Noeud1} , sync , ifconfig eth1 down
33 ${Noeud2} , sync , ifconfig eth1 down
```

Figure-A I-1 Script d'expérience : étude de la performance d'une connexion Ethernet pour des taux de perte de 0%, 5% et 10%.

ANNEXE II

Spécification de la carte Atheros MiniPCI AR5413

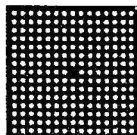
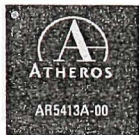
AR5006X

Single chip 802.11a/b/g WLAN Solution



91

ATHEROS®
COMMUNICATIONS



The industry's most highly integrated WLAN solution, enables low-cost 802.11a/b/g products while maintaining range and throughput performance.

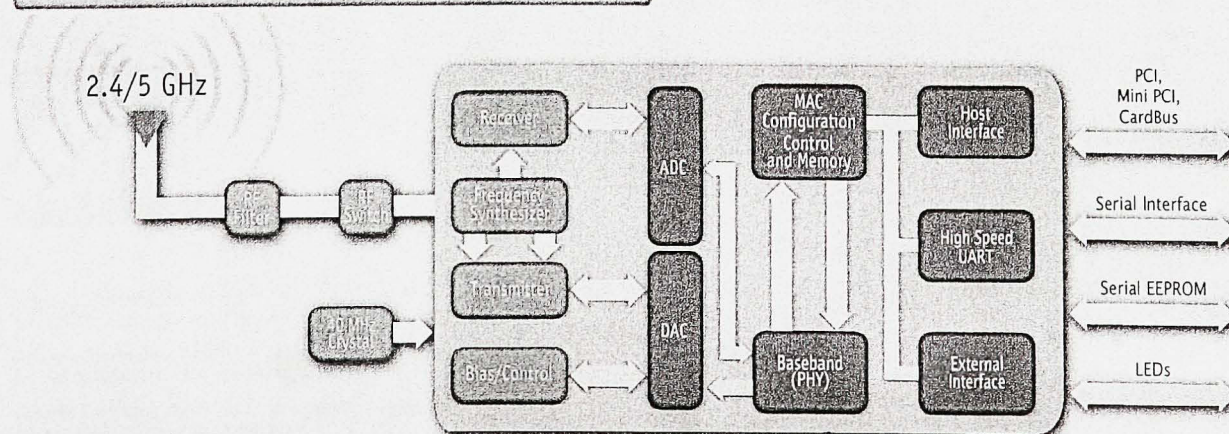
AR5006X Solution Highlights

- Highly integrated single chip CMOS solution with multiprotocol MAC/baseband processor and 2.4/5 GHz radio
- Uses digital CMOS technology exclusively, minimizing power consumption and cost while maximizing reliability
- Support for IEEE 802.11a, 802.11b, 802.11g
- 802.11e standard compatible bursting
- Wireless Multimedia Quality of Service support (QoS)
- Hardware encryption for the Wi-Fi Protected Access (WPA) and IEEE 802.11i security specifications, provides Advanced Encryption Standard (AES), Temporal Key Integrity Protocol (TKIP) and Wired Equivalent Privacy (WEP) without performance degradation
- Extended tuning range (2.300-2.500 & 4.900-5.850 GHz) for worldwide use
- Dynamic Frequency Selection/Transmit Power Control (DFS/TPC) for international operation
- Support for draft IEEE 802.11e, h, i and j standards
- Atheros XR™ eXtended Range technology to give Wi-Fi products twice the range of existing designs
- Power-saving design improvements reduce system power consumption up to 98%

AR5413 Single-Chip CMOS MAC/Baseband/Radio

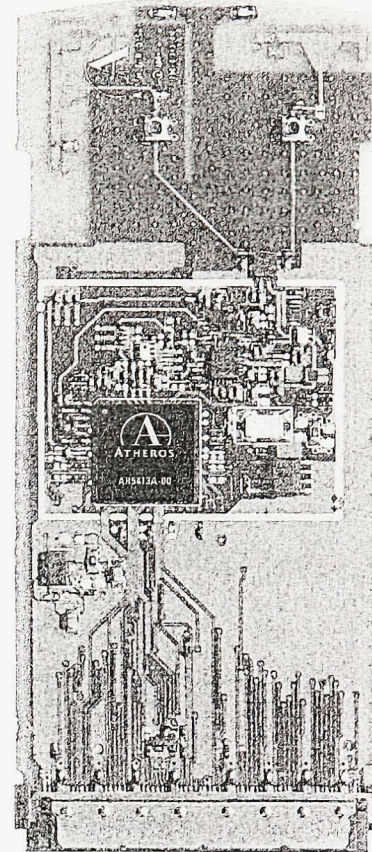
- Support for IEEE 802.11a, 802.11b, 802.11g
- Operates from 4.900 to 5.850 and 2.300 to 2.500 GHz
- Advanced wideband receiver with best path sequencer for better range and multipath resistance than conventional equalizer-based designs
- Integrated low-noise amplifier (LNA)
- External PA and/or LNA can be used for special applications
- Eliminates all IF filters and most RF filters; no external voltage-controlled oscillators (VCOs) or surface acoustic wave (SAW) filters needed
- Enhanced transmit and receive chains
- Atheros XR eXtended Range technology to give Wi-Fi products twice the range of existing designs
- No external FLASH or RAM memory needed
- PCI 2.3 and PC Card 7.1 host interfaces with DMA support
- Integrated analog-to-digital and digital-to-analog converters
- High speed UART with DMA supports data rates up to 1 Mbps
- Serial EEPROM, LEDs, GPIOs peripheral interfaces
- Low power operational and sleep modes

AR5006X WLAN System Architecture



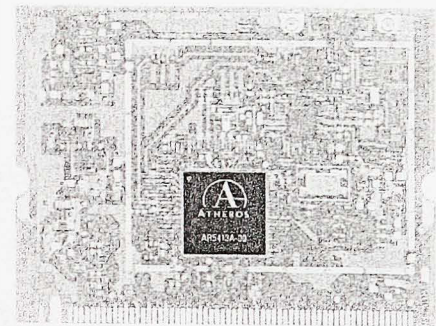
AR5413

AR5006X 802.11a/b/g CardBus Card92



- Windows® drivers for Windows XP, Windows 2000, Windows ME, Windows 98 SE and Windows NT 4.0
- A single driver and firmware code base supports all Atheros chipsets, and provides both backward and forward compatibility with Atheros previous and next-generation multi-standard designs.
- Integrated WPA supplicant supports Windows XP, Windows 2000, Windows ME, Windows 98 SE and Windows NT 4.0
- Client utility supports configuration profiles, current link status, statistics and diagnostics

AR5006X 802.11a/b/g Mini PCI



For more information on Atheros and Atheros WLAN Technology please visit www.atheros.com
Specification subject to change, © 2004 Atheros Communications, all rights reserved.
Atheros, the Atheros logo, Fast Frames, and eXtended Range are trademarks of Atheros Communications, Inc.
All other trademarks mentioned in this document are the property of their respective owners.

AR5006X Solution Specifications

Frequency Band	4.900 to 5.850 GHz and 2.300 to 2.500 GHz
Network Standard	802.11a, 802.11b, 802.11g
Modulation Technology	OFDM with BPSK, QPSK, 16 QAM, 64 QAM; DBPSK, DQPSK, CCK
FEC Coding Rate	1/2, 1/3, 1/4
Hardware Encryption	AES, TKIP, WEP
Quality of Service	802.11e draft
Media Access Technique	CSMA/CA
Host Interface	Mini PCI, CardBus, PCI
Communication Interface	High speed UART
Peripheral Interface	GPIOs, LEDs
Memory Interface	EEPROM
Supported Data Rates	
IEEE 802.11a	6 to 54 Mbps
IEEE 802.11b	1 to 11 Mbps
IEEE 802.11g	1 to 54 Mbps
Chip Specifications	
Chip Specifications	AR5413
Operating Voltage	1.8V +/-5% 3.3V +/-10%
Package Dimensions	13mm x 13mm
Package	224 Plastic Ball Grid Array



Atheros Communications, Inc.
tel: 408-773-5200 fax: 408-773-9940

Atheros Communications, KK – Japan
tel: +81-3-5501-4100 fax: +81-3-5501-4129

Atheros Communications, International LLC – Hong Kong
tel: 852.82061131 fax: 852.82061301

Atheros Communications, International LLC – Taiwan
tel: 886 2 8751 6385 fax: 886 2 8751 6397

ANNEXE III

Statistiques de la plate-forme

Tableau-A III-1 Statistique de développement de la plate-forme : partie serveur

Plate-forme de contrôle d'expérience réseau sur banc de test : serveur de contrôle	
Langage	Java
Nombre de lignes	3533
Nombre de classes	18

Tableau-A III-2 Statistique de développement de la plate-forme : partie nœud

Plate-forme de contrôle d'expérience réseau sur banc de test : gestionnaire de nœud	
Langage	Python
Nombre de lignes	152
Nombre de classes	1
Commandes spécifiques : langage	Script Shell
Commandes spécifiques : nombre de lignes	445
Commandes spécifiques : nombre de commandes	7

ANNEXE IV

Modèle d'abstraction : script d'expérience

```
1 #Paramètres
2 Noeud1=00:0d:b9:15:bc:70
3 Noeud2=00:0d:b9:15:ba:e4
4 Noeud3=00:0d:b9:15:c3:78
5
6 time=600
7 interv=10
8
9 #-----Plateau (up)-----
10 # -> Mise en place du réseau
11 ${Noeud1} , sync , ifconfig eth1 192.168.1.1 up
12 ${Noeud2} , sync , ifconfig eth1 192.168.1.2 up
13 ${Noeud2} , sync , ifconfig eth2 192.168.2.1 up
14 ${Noeud3} , sync , ifconfig eth1 192.168.2.2 up
15
16 # -> Mise en place des routes
17 ${Noeud1} , sync , route add -net 192.168.2.0 -netmask 255.255.255.0 -
host 192.168.1.2
18 ${Noeud3} , sync , route add -net 192.168.1.0 -netmask 255.255.255.0 -
host 192.168.2.1
19
20 #-----Règles (up)-----
21 #Boucle
22 FOR lostRate IN [0,0.005,0.01,0.05,0.10]
23 {
24   # -> Mise en place
25   ${Noeud1} , sync , ipfw eth1 plr ${lostRate} src-ip any dst-ip any
26
27   #-----Déplacement-----
28   # -> Mise en place (up)
29   ${Noeud1} , async , iperf -s , id1
30   ${Noeud3} , async , iperf -s , id2
31
32   # -> Déroulement
33   #Blocs
34   Up ::
35   {
36     ${Noeud1} , async , iperf -c 192.168.2.2 -t ${time} -i ${interv} ,
id3
37     ${Noeud1} , wait , id3
38   }
39   Down ::
40   {
41     ${Noeud3} , async , iperf -c 192.168.1.1 -t ${time} -i ${interv} ,
id4
42     ${Noeud3} , wait , id4
43   }
44   Up-Down ::
45   {
46     ${Noeud1} , async , iperf -c 192.168.2.2 -t ${time} -i ${interv} ,
id5
47     ${Noeud3} , async , iperf -c 192.168.1.1 -t ${time} -i ${interv} ,
```

```

id6
48     ${Noeud1} , wait , id5
49     ${Noeud3} , wait , id6
50 }
51
52 # -> Mise en place (down)
53 ${Noeud1} , kill , id1
54 ${Noeud3} , kill , id2
55
56 #-----Règles (down)-----
57 # -> Mise en place (down)
58 ${Noeud1} , sync , ipfw eth1 destroy
59
60 }
61
62 #-----Plateau (down)-----
63 # -> Mise en place des routes (down)
64 ${Noeud1} , sync , route del -net 192.168.2.0
65 ${Noeud3} , sync , route del -net 192.168.1.0
66
67 # -> Mise en place du réseau (down)
68 ${Noeud1} , sync , ifconfig eth1 down
69 ${Noeud2} , sync , ifconfig eth1 down
70 ${Noeud2} , sync , ifconfig eth2 down
71 ${Noeud3} , sync , ifconfig eth1 down

```

Figure-A IV-1 Script d'expérience : performance d'un trafic TCP au travers d'un routeur rencontrant des pertes.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- Akyildiz I. F. et W. Xudong. 2005. « A survey on wireless mesh networks ». *Communications Magazine*, IEEE, vol. 43, p. 23-30.
- Badarla V., D. Malone et J. Leith. 2008. « Implementing TCP flow-level fairness using 802.11e in a multi-radio mesh testbed ». *Communications Letters*, IEEE, vol. 12, no 4, p. 262-264.
- Balakrishnan H., V. N. Padmanabhan et R. H. Katz. 1997. « The effects of asymmetry on TCP performance ». In *Proceedings of Third ACM/IEEE International Conference on Mobile Computing and Networking 1997 (MobiCom'97)*, 26-30 Sept. 1997. p. 77-89.
- BelAir. 2006. « Capacity of Wireless Mesh Networks ». White paper. 16 p.
- Bicket J., D. Aguayo, S. Biswas et R. Morris. 2005. « Architecture and evaluation of an unplanned 802.11b mesh network ». In *MobiCom 2005: Eleventh Annual International Conference on Mobile Computing and Networking*, August 28, 2005 - September 2, 2005. p. 31-42.
- Camp J., J. Robinson, C. Steger et E. Knightly. 2006. « Measurement driven deployment of a two-tier urban mesh access network ». In *MobiSys2006. The Fourth International Conference on Mobile Systems, Applications and Services*, 19-22 June 2006. p. 96-109.
- Carbone M. et L. Rizzo. 2010. « Dummynet revisited ». *ACM SIGCOMM Computer Communication Review*, vol. 40, no 2, p. 12-20.
- Chen-Mou C., H. Pai-Hsiang, H. T. Kung et D. Vlah. 2006. « WSN07-1: Adjacent Channel Interference in Dual-radio 802.11a Nodes and Its Impact on Multi-hop Networking ». In *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE (Nov. 27 2006-Dec. 1 2006)*. p. 1-6.
- IEEE LAN/MAN Standards Committee. 2007. IEEE 802.11. Piscataway, NJ 08855-1331, Etats Unis: IEEE Standards Association, 1233 p.
- De P., A. Raniwala, S. Sharma et T. Chiueh. 2005. « MiNT: a miniaturized network testbed for mobile wireless research ». In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE (13-17 March 2005)*. Vol. 4, p. 2731-2742.
- Jangeun J. et M. L. Sichitiu. 2003. « Fairness and QoS in multihop wireless networks ». In *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall*, 6-9 Oct. 2003. Vol. 5, p. 2936-40.

- Janghwan L. et Y. Ikjun. 2008. « Achieving throughput fairness in wireless mesh networks based on IEEE 802.11 ». In 2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 29 Sept.-2 Oct. 2008. p. 507-8.
- Karrer R. P., A. Botta et A. Pescap. 2008. « High-speed backhaul networks: myth or reality? ». *Computer Communications*, vol. 31, no Copyright 2008, The Institution of Engineering and Technology, p. 1540-50.
- Kawadia V. et P. R. Kumar. 2005. « Experimental investigations into TCP performance over wireless multihop networks ». In ACM SIGCOMM 2005 Workshops: Conference on Computer Communications, August 22, 2005 - August 26, 2005. p. 29-34.
- Kone V., S. Das, B. Y. Zhao et H. Zheng. 2007. « QUORUM-Quality of service in wireless mesh networks ». *Mobile Networks and Applications*, vol. 12, no Compendex, p. 358-369.
- Koutsonikolas D., J. Dyaberi, P. Garimella, S. Fahmy et Y. C. Hu. 2007. « On TCP throughput and window size in a multihop wireless network testbed ». In 2nd ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, WiNTECH '07, Co-located with the ACM MobiCom 2007 Conference, September 9, 2007 - September 14, 2007. p. 51-58..
- Li-Ping T., S. Wei-Kuan, C. Te-Chung, Y. S. Sun et C. Meng Chang. 2007. « TCP throughput enhancement over wireless mesh networks ». *IEEE Communications Magazine*, vol. 45, no Copyright 2007, The Institution of Engineering and Technology, p. 64-70.
- Peterson L. L. et B. S. Davie. 2000. *Computer networks : a systems approach*. 2ème ed. « The morgan Kaufmann Series in Networking ». San Francisco, Etats Unis, 748 p.
- Ramachandran K., I. Sheriff, E. M. Belding et K. C. Almeroth. 2008. « A Multi-Radio 802.11 Mesh Network Architecture ». *Mobile Networks and Applications*, vol. 13, no Copyright 2010, The Institution of Engineering and Technology, p. 132-46.
- Rosenberg C., L. Jun et A. Girard. 2008. « Engineering wireless mesh networks ». In 2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, 15-18 Sept. 2008. p. 6.
- Weingartner E., H. V. Lehn, K. Wehrle. 2009. « A Performance Comparison of Recent Network Simulators ». *Communications*, 2009. ICC '09. IEEE International Conference on, 14-18 Juin 2009. P. 1-5.

Sites Internet :

Python Software Foundation. 2010. « Python ». <<http://www.python.org>>.

FreeBSD. 2009. « FreeBSD ». <<http://www.freebsdoundation.org>>.

PC Engines. 2010. « PC Engines - Computing on a Human Scale ».
<<http://www.pcengines.ch>>